Edge Hill University

Department of Computer Science

A MODEL TO MANAGE SMART DEVICES IN MOBILE SENSING APPLICATIONS

A thesis submitted in partial fulfilment of the requirements for the degree of Doctor of Philosophy

Supervisory Team:
Professor Nik Bessis
Professor Yannis Korkontzelos

Omosebi, Oladotun Oluwaseyi

Edge Hill University, UK,

May 2021

Dedication

To my parents, Engr. Kunle and Eniola Omosebi, for their blessings and support over the course of the research.

To my wife, Anuoluwapo, for her patience, understanding and emotional support.

To my kids, Jesunifemi and Jadesola, for their inspiring enthusiasm and excitement.

Acknowledgements

I give Honour and Glory to God Almighty for His mercies and faithfulness bestowed upon me throughout the course of the research. I would like to express my sincere gratitude to my primary supervisor, Professor Nik Bessis, who provided immense support and guidance during the time of the study. Your encouragement, clarifications, mentorship, and autonomy on the study gave me immense confidence and learning throughout the study. I would also like to express my utmost gratitude to my second supervisor, Professor Yannis Korkontzelos, for the incessant support during the study. Your guidance, intelligent discussions and strive for excellence in my work motivated me throughout and helped me improve on my understanding of the research process. I would also like to thank Professor Ella Pereira, for the various moments of counsel and advise provided during the study, and for the enabling atmosphere, which allowed me to perform effectively throughout the study. I would like to thank my colleagues as well with whom I shared various intelligent discussions on various subjects over the years, most of which helped me during the course of my study. In some ways, these contributed to understanding my research even better. I especially enjoyed the intelligent discussions with Sarah Michele, whose research shared some semblance with mine. I would also like to extend my gratitude to my parents, Engineer Kunle and Mrs Eniola Omosebi, for their encouragement and support during my study, including their continued assurances over the years. I would like to thank my lovely wife, Anuoluwapo, for her understanding, patience, and assurances while I was away from home for duration of my course. My appreciation also goes to my children, Jesunifemi and Jadesola, whose consistent excitement, inquisitiveness and laughter encouraged me to push further towards the goal of the study. My appreciation also goes to my siblings, Olusola, Olutayo, Mayowa, Taiwo and Kehinde for their support over the years and their encouraging words. I also wish to appreciate my great uncle Adenipekun for his consistent support and encouragement towards maintaining focus on my goal and understanding that it does have an ending.

It is with great appreciation that I wish to collectively thank everyone that has been a part of this journey and to say that this might never have been possible without the participation of every one of you. Thank you and God bless you all.

A Model to Manage Smart Devices in Mobile Sensing Applications

Abstract

The growth in the number and complexity of new smart devices has been exponential in recent years. With the increasing understanding and application of artificial intelligence and machine learning, smart devices have been used in creating new opportunities for intelligent solutions that can enable services suited for smart cities, autonomous systems and ubiquitous systems monitoring and control.

Smart devices, including mobile devices, usually have a small-scale factor and have limited space for batteries, computing, and memory resources. This places a demand for such devices to strictly manage the use of resources to remain in operation for a longer period. In current and upcoming applications of smart devices, such as in the IoT, a network of devices, commonly referred to as a wireless sensor network, needs to gather data by sensing, computing the data, and reporting the information to a base station. Often these data is huge in size and transmitting all the data to the base station would drain the devices of their limited resources. However, the consumption of resources within the device is directly related to the communication and routing algorithm used across the network by each device. Thus, to improve the network's performance through extending its lifetime and addressing more applications than it was specifically built for, the network needs to be sensitive to changes in the context of the application and be able to dynamically select the appropriate routing algorithm to apply based on various performance objectives.

The aim of this research involved the investigation and analysis of the problem, including a study of relevant literature and supporting theory, and culminated in the development of such an adaptive model that can dynamically manage a set of smart mobile devices. It included the investigation of the behaviour of a set of smart devices and their data management approach, while identifying the factors that determined their performance metrics. Metrics considered included energy consumption, bandwidth, and latency. With this knowledge as foundation, an adaptive model with capability to dynamically determine the optimal data management approach in a collection of devices was designed, developed, and evaluated. Various unique single and complex scenarios (scenarios with more than one application running) were used in an evaluation of the model and the results of this process proved that the model outperformed the current state of the art.

Oladotun Omosebi

Date: 25/05/2021

Table of Contents

List of Tables

List of Figures

Publications

Omosebi, O, Bessis, N, Korkontzelos, Y, Pournaras, E, Sun, Q & Sotiriadis, S 2018, Dynamic Scenario-based Selection of Data Aggregation Techniques.. , International Academic, Research, and Industry Association (IARIA) Data Analytics 2018, Athens, Greece, 18/11/18, pp. 27-32.

1 Introduction

1.1 Overview

This chapter provides an introduction and overview of the content of this thesis. It provides a high-level discussion of the underlying concepts, which support the theory and practical aspects of the research. The background section provides the underlying concepts, theory, and a description of the state-of-the-art, which forms a foundation to the main topics of the research. The motivation section discusses the conditions which provide the impetus for the investigation carried out in the research. The research questions indicate the focus of the investigation. Based on these, the aim and objectives describe the tasks that are carried out to address the questions. The contributions highlight the innovations in this research, whereas the structure of the entire thesis is discussed in the later section of the chapter.

1.2 Background

Recent advances in the technology of microelectronic circuits (MEMS) and semiconductor device fabrication have paved the way for the manufacturing of small-scale factor smart devices that can host internally miniaturised computing, memory, and network components (Bakas et al., 2019). The reduced sizes of such devices have enabled the establishment of ubiquitous and pervasive environments, where several smart devices can be grouped together into networks to provide various services, such as group-based sampling and data processing. They are also able to provide services to external applications in various use cases. This trend has enabled the development of intelligent systems such as smart grids, smart meters, smart vehicles, and smart cities (Ganguly et al., 2019; Postránecký and Svítek, 2017; Sun et al., 2016).

Taking smart cities as an example, the underlying goal of such systems includes the creation of smart environments, where shared city resources can be autonomously managed by leveraging data collected by various sensor devices. This can be achieved through the exploration of the installation of an ecosystem of devices to create intelligent smart virtual sensor systems that can autonomously sample phenomena in their environment, process the data internally, and provide actionable reports (Sun et al., 2016). For this structure to be viable, there is need for reliable communication infrastructure, which enables fast unrestricted connectivity and data flows, while also enabling access to the generated data for storage and analysis. This necessary communication channel is already provided by the internet network.

From its inception, the growth rate of the internet has been tremendous over the past few years (Behal et al., 2019). However, more network resources are required to achieve the high-end goals of the imminent IoT. The IoT concept is expected to enable new features for communication between devices and applications, providing a working environment for smarter and more pervasive applications. Sensor devices can detect events and anomalies within their environments and connect to the internet autonomously to provide reports (Atlam et al., 2018). The integration of the internet with cloud-based resources provides enormous potential for such innovative solutions, enabling such sensors to expand their capabilities of sampling, storing and processing more data. Thus, cloud computing enables massive-scale computing by providing platforms based on virtualised resources, which enable parallel processing and integration of data services with scalable data stores (Zhang et al., 2017). The current generation mobile network, 5G, is intended to provide unparalleled fast connectivity between devices, creating a truly ubiquitous network of sensors that can gather data for near real-time control, low latency smart applications, automated factories, analytics, autonomous decision-making, and the provision of various other smart services (Chiang and Zhang, 2016).

The integration and communications between sensor devices and the internet, occurring via the IoT has also recently drawn research interest towards the application of cyber-physical systems (CPSs). Such systems compose of sensors, controllers and actuators, with embedded networking and intelligent components, designed to interface with the physical world and human users, and able to take autonomous decisions based on certain conditions (Sun et al., 2018; Zhang, 2018). This sensor-cloud interaction is expected to increase communications between the IoT, Machine-to-Machine (M2M) communication systems, and the internet cloud, a situation that is expected to lead to the production of high volumes of data, which could easily overload current communication channels (Tseng and Lin, 2018). To address this expected challenge, Big Data techniques could be explored to process the voluminous structured and unstructured data. However, the integration point of application of Big Data in the life-cycle of the data stream would also have a huge impact on the lifetime of the devices in the IoT/M2M platforms, which usually have minimal computing and memory capabilities (Atta et al., 2018; Zhang et al., 2017).

As mentioned, the ecosystem of the IoT enables the deployment of sensors for monitoring various environmental phenomena. The set of sensors communicate via wireless links to share and transmit data to a base station. Equipping the sensors with processing and memory capabilities could be explored in processing collected data locally before transmission. When a group of sensor devices are deployed into a wireless sensor network, their deployment and functioning methods, such as the selected sampling rate, are usually dependent on the purpose for the application. Examples of applications with specific sensing targets include, for example, in natural event monitoring (Alphonsa A. and Ravi G., 2016; Nishikawa et al., 2018; Saputra et al., 2017; Zhu et al., 2012), manufacturing and construction safety (Pievanelli et al., 2013), space exploration (Razfar et al., 2013), structural planning and monitoring (Patil and Patil, 2017), health monitoring (Al Rasyid et al., 2015; Li et al., 2016; Puvaneshwari S and Vijayashaarathi S, 2016), and traffic congestion (Abbas and Yu, 2018), among others. Each of these application scenarios represents a different set of characteristics, which determine the required sampling rate, sensor type and reporting frequency, for example. However, when changes in the environmental characteristics occur within these scenarios, the constraint of the unattended deployment of wireless sensors implies that that the sensors cannot be upgraded, maintained, or reused (Luong et al., 2017).

Considering the advancement and growth of the IoT in recent years, several unknown use cases and scenarios are expected to be established. Based on this fact, there is a clear need for deployed sensors to become more dynamic, adaptive, and self-managed in order to accommodate new scenarios without the need for further setup. This needed dynamic, autonomous and adaptive nature of such sensors would enable them to be more fault-tolerant and be reusable in new applications and scenarios. Thus, a network of sensors, or WSN, consisting of a group of sensor nodes, could then be deployed into any region to capture data, process the data effectively, and communicate such data to a base station, while configuring itself autonomously and performing distributed data aggregation among participating nodes, minimising the use of resources and optimising its performance.

1.3 Motivation

As the IoT continues to expand, there is a tendency for sensor devices to be deployed more often into new environments and applications (Zhang et al., 2019). Now, with the growth in cyber-physical systems capabilities, more data can be generated and stored for analytics in the cloud. Nonetheless, the rate at which the data is being generated will tend towards overloading current network systems (Atta et al., 2018). Thus, recent research has focused on the reduction of the volume of data generated by sensors in the early stage of the data capture or sensing process.

In the usual mode of wireless sensor networks (WSNs), sensors are often deployed into remote or inaccessible regions without needing further maintenance. They are often built for a specific application within a given environment and mostly cannot be reused for a different application. In other words, a group of deployed sensors would be inadequate for use for a different application due to the need for reconfiguration and/or redeployment (Ayaz et al., 2018; Gupta and Quan, 2018; Nishikawa et al., 2018).

With the knowledge that WSNs tend to generate voluminous data, the reduction of such data sizes would tend to require data aggregation approaches to be applied right at the source of data capture or within the network of nodes. Such an approach would need to explore various opportunities for reducing the data size by applying various mathematical functions based on the characteristics of the phenomena. It would also need to be sensitive to the data content based on the spatial and temporal nature of the captured data (Abbas and Yu, 2018; Randhawa and Jain, 2017a). Although, this suits most applications as it reduces the amount of data that is sent to the base station, it does not address the problem of sensor reuse. To facilitate this essential benefit, the working conditions of the sensors need to be dynamically adjusted as well. This would involve modifying some variables, such as the routing algorithm and the logical topology of the sensor nodes, with respect to the running application (Dahda et al., 2018; Daiya et al., 2016; Pournaras and Nikolić, 2017; Sahoo et al., 2017). The logical topology of the sensor nodes in this case implies their communication topology across which a data aggregation technique or routing algorithm runs, which is quite different from the physical layout of the nodes after deployment.

Furthermore, although the described approach seems to address the reuse of the sensor nodes, it does not address the accommodation of new, perhaps autonomously created, scenarios. The establishment of an unknown scenario implies that the sensor nodes are required to interpret the objectives of the scenario, and to self-configure their parameters to adjust to the demands of the scenario. This implies that the sensor network needs to adapt to new scenarios by changing its working parameters to satisfy the requirements of the scenario (Shi and Sha, 2019).

Based on the above discussion, it is obvious that there is immense benefit in the deployment of a WSN into a scenario, where the network is able to manage itself by autonomously detecting the environment of its application, building its logical topology, selecting its preferred routing algorithm, sampling rate and data transmission approach. This also makes it adaptive to changing condition changes in its environment, which require it to make changes to its parameters to optimise its performance. Achieving this requires a dynamic selection of the right data aggregation technique, which would optimise the network's working conditions given the running application's set of criteria and priorities (Al-Tabbakh, 2017).

This research is focused on the design and development of a model that is used to manage a set of sensor devices in a wireless sensor network, to perform optimally within a given scenario, while at the same time, being capable of adapting to a new configuration, based on changing environmental conditions. The underlying goal of this set of dynamic configurations is to select the most appropriate data aggregation technique which determines the best logical topology and optimises the network metrics.

The process to develop this model will involve the study of literature covering various facets of wireless sensor networks, while taking into account the application of experiments to determine relationships between many variables. Data aggregation techniques shall be studied in detail, while their determinant contrasting parameters will be identified. Eventually, technique and scenario data is be used to determine relationships and develop machine learning models that can predict applicable techniques given new event parameters.

The remaining sections of these chapter shall discuss research questions, aims and objectives, contribution, and the structure of the thesis.

1.4 Research Question

The main research question is as follows:

Can we design and develop a robust and efficient model that can dynamically manage sensing and computational tasks among a network of sensors, while being adaptive to changes within the environment at the same time?

1.5 Aim and Objectives

To address the aforementioned research question, the following aim is formulated:

To design, develop, and evaluate an intelligent and adaptive model that can dynamically manage a set of sensor devices within a wireless sensor network. Based on a set of criteria and priorities, the model will be able to optimise data aggregation within such sensor networks, by enabling adaptation, context-awareness, self-discovery, and self-configuration.

The following objectives are essential to achieve the above aim:

1. Perform a critical literature review to determine supporting discussion on the relationship between resource characteristics and sensing scenarios. This objective is looking whether a set of measurable parameters can be identified to model the resource characteristics of different wireless sensor scenarios.
2. Design and develop models to represent sensor networks, where the characteristics of running application can be correlated with the wireless sensor network. This objective is looking whether the characteristics of a wireless sensor network can be correlated with the characteristics of the application running on the network.
3. Develop models for data aggregation techniques to be applied in complex wireless sensor network scenarios to evaluate their performance. This objective is looking whether different data aggregation techniques can be considered optimal for specific application scenarios.
4. Perform an evaluation of the impact to network performance of making changes to various wireless sensor network variables. This objective is looking whether the parameters of a

wireless sensor network can be dynamically adjusted to perform optimally based on the running application.

5. Develop an adaptive model to predict the best data aggregation technique based on the characteristics of a running application in a wireless sensor network. This objective is looking whether a sensor network can adapt autonomously by changing its own set of working parameters based on a new scenario or application in order to optimise its performance.

1.6 Contributions

This study shall provide the following contributions:

1. Detection Needs Analysis Model
   This includes analysis, design, and establishment of various WSN attributes and relationships that were used to develop the intelligent model in this study.
2. Intelligent Dynamic and Adaptive Model
   An adaptive intelligent model was developed, which was able to determine the best data aggregation technique given simple and complex WSN scenarios. This capability enabled constrained sensors and mobile devices to improve their performance under the limitation of their resources. The model enabled the determination of the right optimal values for the network's parameters, which enabled the wireless sensor network to utilise the appropriate data aggregation technique that fitted the context.
3. Prototype Framework and Source Code
   A framework and source code that could be used in future research to model new data aggregation techniques.
4. Experimental Conclusions
   The conclusions reached based on the study also provide a reference guide to further study in the area of WSN.

1.7 Thesis Organisational Structure

The structure of the thesis is described as follows. Chapter 2 involves a study of literature, which provides the theory underlying the research study. The chapter covers past and current studies, which support the motivation and goal of the study, as well as highlighting specific hypotheses and theory that relate to the study. Chapter 3 discusses the methodology used to in the research, and includes data gathering, tools and proposed simulations. Chapter 4 covers the design and provides a needs analysis, which includes identification of concepts and components that are required to achieve the objectives. It also presents details on WSN dimensions, the workflow of typical scenarios, and explores the relationship between the components, which make up the WSN application. Chapter 5 covers the development of the machine learning model and discusses the relevant relationships between various WSN components. It also presents a reference architecture and performs a formal analysis of the concepts used throughout the study. It introduces the performance metrics used in the study and provides details on the attributes used to generate and process data for the study. Chapter 6 discusses the implementation of the machine learning model and software prototype. It also discusses the hardware and software environment, and the challenges encountered. Chapter 7 discusses the testing and evaluation of the intelligent model. The discussion includes analysis of the dataset, the intelligent model training and testing, accuracy plots, evaluation using various scenarios, The chapter also discusses the model evaluation based on various complex (more than one) scenarios and evaluates these with reference to the state of the art. Chapter 8 presents the conclusions of the study, discussing the results, the research outcomes and the contributions. It also highlights the limitations and the recommendations for the further research based on the study.

2 Literature

2.1 Overview

This chapter presents the literature review of this study. It discusses sensor devices, their evolution, architecture and application scenarios, including their use in the Internet of Things (IoT) scenarios are also discussed. The integration of sensors with recent cyber-physical systems are explored and the growing challenge with data collection and processing are also discussed. The idea of a set of sensors in a network, subsequently referred to as a wireless sensor network, is also mentioned. The problem involving the impending overload of data due to continuous collection, which is driven by the reducing cost of storage, and how this challenge could be addressed, are discussed. The need for data aggregation within wireless sensor networks is highlighted and various opportunities to achieving this are explored. The application of data aggregation techniques is introduced and discussed, while emphasise on the relationship between their innate attributes highlighted. The need for the dynamic selection of data aggregation techniques within the context of a dynamic IoT are discussed. This discussion extends to the necessity for an adaptive model, which can select appropriate techniques. The chapter ends with a discussion on supporting literature for an adaptive, intelligent model, which can detect the context of an event, and based on this knowledge, be able to select the right technique to optimise the performance of the WSN.

2.2 Internet of Things and Wireless Sensor Networks

The IoT is composed of collections of smart devices, inter-operating to serve one or more high-level application requirements (Alnahdi and Liu, 2017). The sensed data can be used for various purposes based on the requirements of the running application, such as is applicable in vehicle control, factory floor automation, equipment management, and in machine learning for weather prediction (Deloitte, 2018; Ghosh et al., 2019). The increasing number of devices in the IoT is expected to generate high volumes of data, having inherent characteristics such as high volume, veracity, velocity and variety, and requiring relevant Big Data techniques for processing (Benjelloun et al., 2015; Boubiche et al., 2018; Sun et al., 2016). Likewise, new sensor devices are being produced with smaller form factors, enabling application deployments at wider scale and lower cost. The demerit of smaller size devices includes reduced-size battery packages and a constraint on available power sources, which essentially minimizing resources available for normal operation (Jyothi and Cholli, 2019). Thus, resource utilization needs to be managed effectively to minimize energy consumption during operation, and thus, maximize network lifetime.

A typical node (or sensor device) , which samples and generates data, could also be referred to as a source node. The last node in the network, which receives the data from the group of network nodes is usually referred to as the sink node. The node could also be tasked with submitting requests into the network, and to receive the aggregated result from the network. In order to provide a useful function, nodes are usually deployed in groups in the form of a network, usually referred to as a wireless sensor network (WSN). The objectives of such a network includes tasks such as monitoring and detection of physical phenomena (Juneja and Das, 2019). They are used in various practical applications such as smart environments (Alnahdi and Liu, 2017), smart home and service monitoring (Alnahdi and Liu, 2017; Salunke and Kate, 2017), health (Chatterjee et al., 2017), farming (Baldovino et al., 2018), and natural event monitoring (Galappaththi and Weerasuriya, 2018; Ozbey et al., 2018), etc.

While applications are running in a WSN, the WSN exhibits certain characteristics, which can be correlated with the requirements of the application. This research includes these characteristics to categorise WSN applications while correlating them with data aggregation techniques. Some of these characteristics are discussed below (Djedouboum et al., 2018; Jyothi and Cholli, 2019):

1. Network Topology: this defines the structure of the network of sensors. It refers to the data routing approach or algorithm used within the network by the nodes during communication. It could fall into one of the following options: Star topology, Mesh topology, Tree topology, Cluster topology, Hybrid Topology.
2. Node Similarity: This identifies the similarity in the capabilities of the sensor nodes. Nodes with similar capabilities are described as homogenous, or else they are referred to as heterogenous. These are described below:
    a. Homogenous (network): consists of nodes with similar resource capabilities in computing, networking, and memory. Also, they usually target the same type of phenomena. As an example, weather forecasting would utilize sensors with similar settings in a selected area in order to obtain reliable and consistent results.
    b. Heterogenous (network): consists of nodes with varying component capabilities. They could measure the same set of phenomena, but at different resolutions due to their different resource capabilities. This configuration also introduces settings where nodes could hold super-node status to coordinate data aggregation tasks. Heterogeneity can be further classified into various types. These include *Node-based* (sub-classified under Layer-based, Hardware-based, Sensor/Actuator-based, and management-based), and *Network-based* (sub-classified under Topology-based, Location-based, and Working Model-based (Yıldırım and Tatar, 2017).

3. Communication Mode: this specifies the approach used to send reports from all other nodes to the sink node and could be one of the following:
    a. One-Hop Communication: nodes transmit directly to the sink node, usually via long distances if the sink node is placed far from the network. This tends to increase energy consumption for nodes which are far from the sink node.
    b. Multi-Hop Communication: nodes transmit via intermediate nodes to get move to the sink node. This tends to reduce transmission distance and thus, minimize energy consumption. A relevant network architecture is required in this case and could be one of the following:
        i. Flat architecture - where no specific structure exists among nodes, or,
        ii. Hierarchical architecture – where communication hierarchies are relied on for communication.

    This research mostly focuses on the multi-hop hierarchical architecture.
4. Routing Protocol: this determines the communication mode between nodes within the network. It could fall into one of the following three classes (Mehta and Saxena, 2018):
    a. Data Centric Protocols: such protocols label sensed attributes and use these to route the data across the network. Examples include Directed Diffusion (DD), Sensor Protocol for Information via Negotiation (SPIN), etc.
    b. Hierarchical Protocols: such protocols attempt to minimize energy consumption by leveraging multi-hop communication between nodes. Examples include Low Energy Adaptive Clustering Hierarchy (LEACH), and Power Efficient Adaptive Clustering Hierarchy (PEACH), etc.
    c. Location-based Protocols: such protocols rely on the location of nodes in order to coordinate data transmission in the network, as well as manage the impact of communication on overall network resources. Examples include Geographic Adaptive Fidelity (GAF), Minimum Energy Communication Network (MECN), etc.

    This research mostly focuses on the hierarchical routing protocol.

The phenomena being monitored by a WSN usually has certain inherent characteristics, which can be used to determine the necessary setup for an applicable wireless sensor network (Punniamoorthy et al., 2018). For instance, in a personal area network, the application environment dictates the topology. This is because the sensed data will typically be streamed to a central node placed somewhere on the body, a strategy best met with use of a star topology (Jung et al., 2016).

In normal operation, a WSN is expected to the generate huge amounts of data. The data includes features related to WSN characteristics (or attributes), which can be used to evaluate the WSN's performance (Boubiche et al., 2018; Djedouboum et al., 2018). In other words, they can be used as performance metrics to evaluate the performance of data aggregation methods currently running in the network (Sasirekha and Swamynathan, 2017). The following list consists of candidate metrics (also referred to as objective functions), which could be used to evaluate the performance of a data aggregation technique in a running WSN. These metrics were considered important for the study since they could be used to measure the performance of the WSN when a technique is running within it. Such measurements enable the determination of the best technique for the selected WSN, and subsequently the data to model the technique's behaviour. The metrics selected for use in the study were dependent on the capability of the NS3 simulator as well in terms of what variables in the WSN could be measured:

1. Energy Consumption: this represents the sum of energy consumption across all nodes within the network. Since sensor radio reception and transmission are considered the top energy consuming activities of nodes (Rosadi and Sakti, 2017), the attributes of amount and distance of transmissions need to be minimised in order to minimise energy consumption. This was considered an important metric for the study since the primary objective of data aggregation in WSNs is the minimisation of energy consumption. Thus, it is used in the evaluation of WSNs and DATs in later chapters. Its unit is in Joules (J).
2. Latency: this is the time duration from when a request is sent into the network from the sink node, to when a report returns the sink node. This is impacted by various attributes, such as volume of data transfer, and the number of participating nodes. This metric is also considered as an important metric in the study since it represents an obvious differentiating factor in the performance of DATs. It is thus included in the metrics used to evaluate WSNs and DATs. Its unit is in nanoseconds (ns).
3. Network Lifetime: this refers to the timespan during which the WSN is capable of effectively performing the objectives of the running application. This is based on the WSN scenario, where the appropriate definition for Network Lifetime needs to be defined. For instance, it could imply using when the first node dies (FND), or percentage of nodes die, and also, last node dies (LND). This metric is directly related to the energy consumption since a minimisation of energy consumption directly leads to an extended network lifetime. It is, thus, considered an important metric. However, it is not represented with a variable in the study. Its unit is in seconds (s).

4. Bandwidth Utilization: this represents the total bytes used by the network in topology setup, through sensing, and to the transmission of data to the sink, and is defined by the attribute – number of bytes. The more data is used within these stages, the higher the attribute value, and the higher the objective function rises. This also directly correlates with the higher consumption of energy since each byte requires a minimum amount of energy for transmission. This metric was also considered important for the study. It was measurable with the simulation tool and its value could be directly related to the volume of data transmitted across the WSN. This it was used further in the study. Its unit is bytes (B).
5. Scalability: this represents the capability of the network to scale up under demand, while maintaining an expected level of quality of service. The term "scale-up" could imply an increase in the number of nodes or the network sampling rate. This could occur if the field size of the application event increases, or the number nodes grows due to new deployment, or more nodes turning themselves online due to increased demand. This metric was not used further in the study for various reasons. Its computation was considered more complex, that the first four metrics since it required further modelling of the network size and real-time spread. It was also discarded to manage the scope of the study.
6. Fault Tolerance: this represents the capability of the WSN to maintain reliable performance in the eventual failure of nodes across the network. Such failure could be due to depleted power supplies, isolated nodes, or obstructions to data transmission from node to node. Other reasons are possible due to the unpredictable nature of the context of a WSN application. The same explanation applies to this metric as does for scalability. It was not used further in the study.

The above list consists of WSN metrics, which are measured across the network during an application's lifetime. Since the values of these functions change dynamically based on the network structure and routing algorithm applied across the WSN, they are used in this research to evaluate the performance of data aggregation methods used in the network. Given the context of an application, and the varieties of algorithms that can be used, the outcome of these variables will tend to differ for every data aggregation approach.

2.3 Wireless Sensor Networks and Data Aggregation

To address the challenge of managing high data volumes within WSNs, various data aggregation methods are usually applied. Within the context of a WSN, data aggregation can be described as a distributed processing approach to data spread across several sensor nodes, based on a set of rules (Boubiche et al., 2018; Zhang et al., 2018). It involves a decentralized computation of various network attributes that can be consumed by running applications. The characteristics of spatial and temporal correlation within the sensed data provides an opportunity for applying this method, as it enables the detection and aggregation of duplicates or near-duplicates based on certain rules (Yang, 2017). Real-life scenarios where this occurs include carbon monoxide monitoring, and the detection and monitoring of seismic vibration in earthquakes. In these applications, sensors in proximity could hold consistent values across a wide region, requiring some pre-processing to summarize the data.

Data aggregation techniques (DATs) enable coordination among the sensor nodes by providing a logical structure to support the communication between nodes. While specifying the routing topology for the network, DATs need to establish a balance between the various WSN performance metrics as mentioned in section 2.2, such as energy consumption and network lifetime. For optimum performance, the chosen DAT for a WSN needs to be closely aligned with the running application on the WSN (Boubiche et al., 2018). The combination of the application context and the WSN could be interchangeably referred to as a *scenario* (AlMansour and Alahmadi, 2018; Boubiche et al., 2018). The stages of data aggregation are shown in figure 2.1. The figure indicates that data is gathered at the sensor nodes, and then aggregated based using one of two options. In the first case, the data is aggregated as it is transmitted via intermediary nodes to the base station. In the second case, the data is not aggregated until it reaches base station.

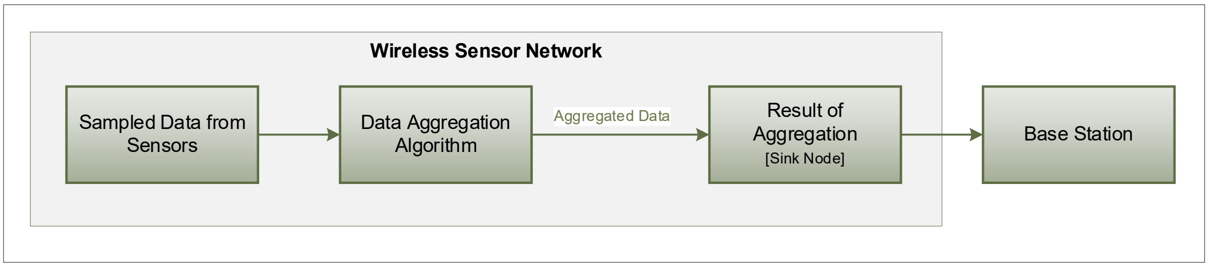

Figure 2.1 - Data Aggregation Process - composed from (Randhawa and Jain, 2017a))

2.4 Classification of Data Aggregation Techniques

Data aggregation techniques can be classified based on certain attributes. The attributes have the same application across all techniques. The following discussion covers these attributes. The attributes are important for this study since they represent measurable variables, which can be used to estimate the performance of techniques, and thus, providing the capability to compare such performance:

- Network Architecture and Topology: this refers to the collection of sensor nodes, which form a network of sensors. The physical distribution and placement is referred to as the network architecture, and captures variables such as the distance between nodes and communication interference. The network topology refers to the data communication paths between the nodes. A data aggregation technique determines a specific network topology, usually based on the network architecture, which enables data routing and aggregation.
- Routing protocol – this relates to the selected technique and topology and includes protocols which are described as chain-based, tree-based, and cluster-based.
- Objective Goal – this defines the specific purpose of the application and relates directly to the WSN's performance metrics. For example, an application could include an objective of minimising latency or maximising network lifetime. This association enables the performance of a technique used within a WSN to be evaluated.
- Optimizing parameters – represents WSN characteristics (or attributes), which have an impact on the response of the applied DAT. Changes to these variables tend to affect the performance of the DAT during the lifespan of the application. Examples of such attributes include node count, field size, and sampling rate. While these attributes could be linked to the application, the network, or the overall WSN, DATs perform differently given a different vector of a set of values.

Due to the huge number of data aggregation techniques available for use in WSNs, many instances of attempts to categorise techniques was discovered in literature. Table 2.1 documents many of these cases where a specific set of dimensions were used in an attempt to categorise various techniques. This is important for this study as there is a need to categorise techniques to distinguish them based on their behaviour and performance.

Table 2-1 - Example classification approaches for data aggregation techniques

No	Source	Focus of Classification	Method of Classification
1	A Survey of Distributed Data Aggregation Algorithms (Jesus et al., 2015)	Data Aggregation in WSNs	• Function Types - duplicate sensitive/insensitive, • Communication • Routing and network (structured-hierarchical, unstructured-flooding/broadcast, etc.), • Computation (e.g. decomposable averaging/sketches, etc.)
2	Data Aggregation in Wireless Sensor Networks: Previous Research, Current Status and Future Directions	Data Compression in WSNs	• Topology type • Technique Objective.

	(Randhawa and Jain, 2017a)		
3	Practical data compression in wireless sensor networks: A survey (Srisooksai et al., 2012)	Data Compression in WSNs	• Data compression by energy efficiency
4	A Taxonomy of Wireless Micro-Sensor Network Models (Tilak et al., 2002)	Communication functions in WSNs	• Communication functions • Data delivery models • Network dynamics with respect to power demand
5	Issues of Data Aggregation Methods in Wireless Sensor Network: A Survey (Sirsikar and Anavatti, 2015)	Data aggregation in WSNs	• Strategy • Delay • Redundancy • Accuracy • Energy consumption • Traffic load
6	Data-aggregation techniques in sensor networks: A survey (Rajagopalan and Varshney, 2006)	Data aggregation in WSNs	• Network lifetime • Latency • Data accuracy • Security
7	A survey on sensor networks (Akyildiz et al., 2002)	Wireless Sensor Networks	• Protocol layer

Research initiatives to create new data aggregation techniques usually commence by selecting the appropriate network architecture, or topology, such as cluster, star, tree, or mesh topologies. The following topology types are discussed below. All the topologies mentioned below are used later in the study since they are are fundamental characteristics of the discussed techniques.

1. Cluster Topology: this topology involves an organisation of nodes in cluster groups, with each cluster having a node that represents the cluster head. Other nodes within the network must then select the closest cluster head to them as their next communicating node and join a cluster as cluster member, with other nodes connecting to the same cluster head. The function of the cluster head involves receiving data from the cluster members and aggregating these for further transfer to the sink node. The transfer is done via direct transmission or via multi-hop routes provided by other cluster heads.
2. Tree Topology: this topology involves an arrangement of nodes in a tree structure, where a node represents the root node, some nodes maintain edge positions as the sensors, while other nodes serve as intermediary nodes. The tree is usually built using one of various tree construction algorithms such as Kruskal’s minimum spanning tree algorithm (MST). Leaf

nodes trigger off the sensing task, sending data to their parent intermediary node. While data is being transmitted, data aggregation is taking place to reduce the data received before transmitting to the next node in the hierarchy. The intermediary nodes continue to pass the data on until it reaches the sink node, which forwards it to the base station.

3. Chain Topology: this involves an arrangement of nodes in a chain-like structure. The chain consists of the end nodes, which are at both ends of the chain, the intermediary nodes, which receive data from the end nodes and pass on to the head, and the chain head node, which is responsible for sending the aggregated data to the base station. The sensing starts at the end nodes and continues to the chain head, while data aggregation occurs as the data passes along the chain.
4. Mesh Topology: this involves a lack of any form of arrangement of the nodes in the network. Thus, nodes connect to other nodes in their proximity based on no rule or method. This structure provides various multi-links for data routing and maximizes fault tolerance within a WSN.

Figure 2.2 provides a hierarchical classification of data aggregation techniques within the context of WSNs. The three main approaches to data aggregation shown in the figure include centralized, in-network and hierarchical.

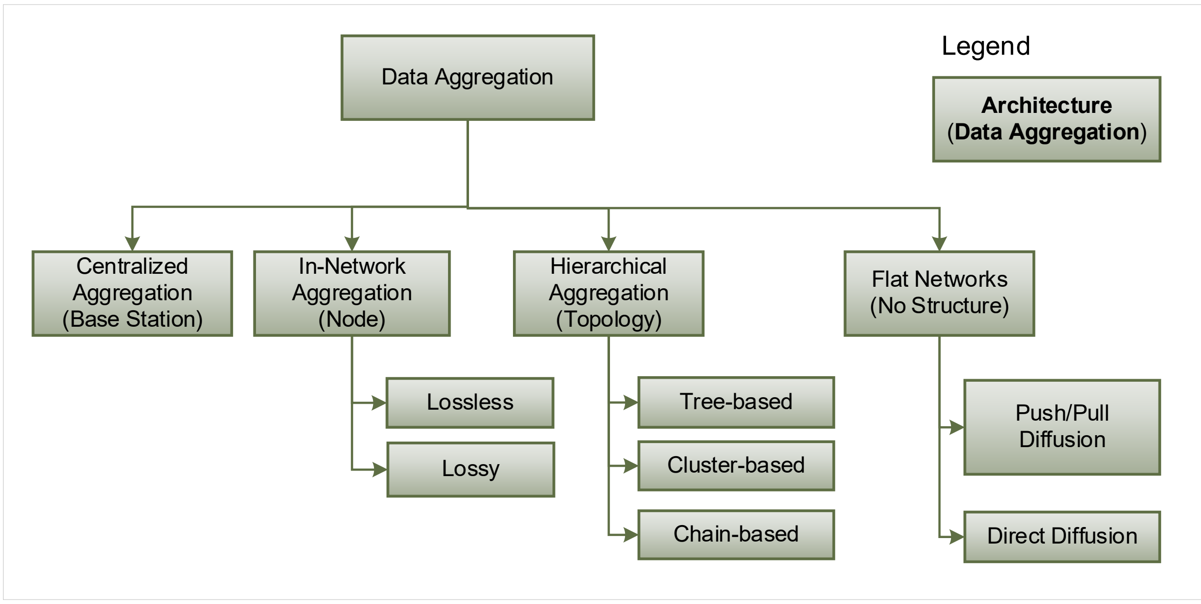

Figure 2.2 – Classification of Data Aggregation Techniques, composed from (Al-Doghman et al., 2017; Hiteshreddy et al., 2015; Jesus et al., 2015; Randhawa and Jain, 2017a; Zhang et al., 2018)

Centralized aggregation implies that data aggregation occurs at the sink node. All network nodes sample their environment and send all the data to the sink node, by single or multi-hop communication, where final data aggregation is performed. In the *In-Network* form of data aggregation, the intermediate nodes are responsible for performing internal aggregation of data. This applies to the data captured by each node, before being added to data received from other nodes (Zhang et al., 2018). This form of aggregation could be sub-divided into lossless and lossy aggregation, based on how the packet size is managed. Lossy aggregation involves the use of one or more mathematical functions, such as SUM, COUNT, MAX, MIN, and AVE, to compress the packet size. This is essential when the volume of data is larger than the capacity of the receiving node. In lossless aggregation, all generated packets are sent towards the sink node. The more popular *Hierarchical* aggregation implies that data aggregation is performed based on a hierarchical arrangement of nodes using one of the available topologies: Tree, Cluster, and Chain (Randhawa and Jain, 2017a). Thus, as data is being transmitted across the network, data aggregation is being performed to summarize the data before it reaches the sink node.

Data aggregation techniques also have various static characteristics, also referred to as attributes. Such attributes inherently affect their behaviour and are immutable for the technique, even during the lifespan of the scenario (WSN and application). Examples include location awareness (of nodes), homogenous vs heterogenous nodes, algorithm types used, etc. The choice of homogenous vs heterogenous nodes implies that a technique works mainly on a certain configuration of nodes, such as nodes with similar components (homogenous), or otherwise (heterogenous). A technique could also require nodes to be location-aware (e.g. PEGASIS), or non-location-aware (e.g. LEACH, HEED, etc). These attributes are considered important distinguishing variables for determining and identifying the behaviour of the DAT while being active in a scenario. However, since these attributes are static and inherent in the subsequent behaviour of the technique, they might not be considered relevant if the behaviour of the technique can be accessed via their performance given specific network configurations. This also relates to other technique attributes, which are computed, and relate to the behaviour of the DAT. Such attributes are computed based on other attributes tied to either the WSN or characteristics in the application. Such attributes include, for example, the active number of nodes in the network, the average distance between nodes, or the (instantaneous) sampling rate of the application (Prabha et al., 2018).

2.5 Types of Data Aggregation Techniques

Various data aggregation techniques have been proposed over the years. As mentioned in the last chapter, while they are mostly classified based on their topology, an important characteristic also involves their target working environments, which are also related to the WSN application (Dhand and Tyagi, 2016; Jesus et al., 2015; Randhawa and Jain, 2017a). To discuss some of the characteristics further, the following sections will cover various example DATs according to their topologies and target network setup. Using these examples, the distinctions between DATs are highlighted, their application-specific nature discussed, while the need to apply different DATs during the lifetime of a WSN scenario is emphasized.

A data aggregation technique in a typical WSN determines the data computation and routing algorithm for nodes within the WSN. This task is inherently constrained by the need to optimise certain objective functions, as defined by the WSN application's criteria. Various techniques typically target specific primary functions for optimization. For instance, some techniques target the minimisation of energy consumption. Some of these include:

- Hybrid Energy-Efficient Distributed (HEED) (Younis and Fahmy, 2004a),
- Power Efficient Gathering in Sensor Information Systems (PEGASIS) (Lindsey and Raghavendra, 2002),
- Adaptive Energy Aware Data Aggregation Tree (AEDT) (Virmani et al., 2013),
- Energy-based Data Aggregation (EBDA) (Huang and Zheng, 2012),
- Dynamic Balanced Spanning Tree (DBST) (Avid Avokh and Mirjalily, 2010),
- Low Latency Diffusion of Information (Intanagonwiwat et al., 2003),
- Energy Efficient Adaptive Data Aggregation using Network Coding (ADANC)

The primary techniques discussed in following sections include LEACH, HEED, PEGASIS, AEDT, and DBST.

2.5.1 Cluster Topology-based Techniques

Cluster topologies arrange nodes in clusters, with a common cluster head per cluster of nodes. The topology is usually built to optimise scalability, minimise energy consumption, and thereby extend lifetime. However, data accuracy could suffer as the number of the nodes in a cluster increase (Jyothi and Cholli, 2019).

LEACH

The LEACH technique is considered a classical representation of a WSN data aggregation technique. Being among the first techniques developed, it often serves as a primary reference point for evaluating new techniques (Handy et al., 2002; Sasirekha and Swamynathan, 2017). LEACH uses a cluster topology, where clusters of nodes are created within the network. It works based on a process consisting of two stages – Setup, and Steady State, The processes are described using the following algorithm:

Setup Stage

i. the technique selects the network head nodes by applying its chosen algorithm

ii. the technique builds the network topology using the heads, and applies algorithm to select nodes, e.g., node selects closest head node as its next communicating node

iii. chosen head nodes are notified by nodes connected to them

iv. node heads allocate time slots for communication to all connecting nodes

Steady State Stage:

v. sensor nodes start to sense their environment based on the purpose of the WSN application

vi. sensor nodes pass captured data and send to head node, or the next adjacent node

vii. head node or next adjacent node sends node onto the next node, or to the base station if the last head node is reached.

viii. steady state stage is repeated until the energy level of the WSN is insufficient to run the application effectively.

As described in the above algorithm, the completion of the second stage of the LEACH DAT signifies the end of a process involving topology setup and transmission, which is similar for many other DATs. The process take place in a timespan usually referred to as a *Round* (Darabkh et al., 2017; Raghunandan et al., 2017). It is noteworthy that virtually all techniques understudied during this research had a similar process. However, the termination of a round could be different for various techniques. The concept of "round" can however, be considered as one of the yardsticks for evaluating the performance of DAT, and is thus, used in this research.

The approach used in LEACH to select the cluster head involves a probabilistic random function which considers the active nodes that are yet to become cluster heads after a given number of rounds. A round within the scope of this technique involves the determination of cluster hears, the formation of clusters, the sensing, and the transmission of data. This is a special requirement for the LEACH DAT. Although, the goal of LEACH involves the minimisation of energy consumption, it does not consider the remaining energy of nodes when selecting new cluster heads. The originally designed LEACH DAT defines that all cluster heads should send the aggregated data results directly to the sink node. However, other research focused on improving its performance have redefined the approach to require that cluster heads use a multi-hop technique to transmit data to the sink node (Bongale et al., 2017; Gantassi et al., 2017; Yassein et al., 2017). An additional requirement of the LEACH technique includes the constraint of active number of heads to 5% of all nodes in the network. Variants of the LEACH DAT have been developed by numerous researchers (Yassein et al., 2017), such as LEACH-EA (Gantassi et al., 2017), and LEACH-C (Tripathi et al., 2013), with similar research taking place for other DATs as well (Darabkh et al., 2017).

Similar to the algorithm of LEACH, various other techniques use random head selection as a strategy, however combining the attribute of node remaining energy as well. Such techniques, commonly referred to as energy-efficient algorithms, include DEEC (Qing et al., 2006), and EEHE (Kumar et al., 2009). The DEEC DAT uses a probability function based on the ratio of the residual energy of each node, to the average remaining energy of the network. Nodes are split into normal and advanced nodes, where advanced nodes have α times more energy than normal nodes. Thus, probability P is calculated differently for each node, based on the group it falls into. The EEHE DAT assumes a three-level heterogenous network, where each level contains nodes based on their energy level.

HEED

The HEED DAT uses a similar algorithm to LEACH but differs in the approach to cluster head assignment. Instead of using a random function, it considers the remaining energy of the nodes to selected candidates for the cluster head role. It also considers node density, formerly defined as the ratio of the number of active nodes within a node's given radius, to the number of active nodes across the network (Chand et al., 2014; Pillai and Jain, 2018). HEED is a homogenous DAT, relying on a consistent similarity of node components and capabilities. The concept of a DAT's *Assumption* can be used to define this expectation on node characteristics, a behaviour shared by other DATs as well. It is revisited further in the study as a yardstick for differentiating DATs. To illustrate further, when a DAT requires location-awareness to operate effectively, it will expect, or "assume" that the WSN nodes have location-awareness capabilities. However, HEED does not assume (or require) location awareness in the network nodes.

There are other variants of the HEED DAT, which use the primary algorithm of HEED with a few adjustments to attain higher efficiency. Integrated HEED (iHEED) (Younis and Fahmy, 2005), for instance, applies the functions AVE and MAX in integrated data aggregation during multi-hop routing. Other variants, such as hetHEED-1, hetHEED-2, and hetHEED-3, represent mainly heterogenous versions of HEED with nodes distributed across, 1, 2, and 3 levels, based on their remaining energy levels. In addition to using the same attributes as used in HEED's computation, they also include the distance between a node and the sink. The values of these attributes are then applied to fuzzy logic to obtain the probability that is used to determine which nodes qualify as cluster heads (Chand et al., 2014).

2.5.2 Tree Topology-based Techniques

Data aggregation techniques that rely on the tree topology create logical links between nodes based on a tree structure. Nodes are assigned roles such as root, intermediary and leaf. The leaf nodes are usually at the tail end of the tree and often trigger-off sampling across the network. Their data is sent to intermediary nodes, which then pass on the data to the next nodes in the tree hierarchy, towards the root node. Data aggregation takes place across all the nodes, and up till the data gets to the root node (or sink node), which forwards the results to the base station. Important tree constructing algorithms to mention include Kruskal's Minimum Energy Spanning Tree (MEST), which computes routes based on the remaining energy of a node, and Dijkstra's Shortest Path Spanning Tree (SPT), which computes routes based on the shortest distance to the sink node. These, and several other algorithms, have different impacts on the behaviour of a DAT and can be used as a yardstick to distinguish them (A. Avokh and Mirjalily, 2010).

AEDT

The AEDT DAT (Adaptive Energy Aware Data Aggregation Tree) uses a tree topology for its logical routing network (Virmani et al., 2013). The root node is selected based on the maximum available energy of the node, while communications between two nodes is controlled based on the available buffer at the receiving node. It also applies a common function of keeping only interacting nodes awake, based on a TDMA (Time-division multiple access) protocol for network communication. All successful communication paths are stored in a memory table, and this is used in future communication tasks. Its algorithm also uses the *Shortest Path First* method for its tree topology construction.

The first step taken by AEDT involves the selection of a parent node. Each node first broadcasts it's energy level E_{avail} across the network. Based on a distributed algorithm, the node with the highest energy level is selected as the parent node. The selected node then broadcasts its status to the entire network. The available energy for a node is calculated using the following formula:

$$E_{avail} = E_b\,(t_2) - E_b(t_2) + \int_{t_1}^{t_2} P_c(t) \qquad - \qquad \text{eq2}$$

Where,

P_c implies network power consumption (Watt)

$E_b(t_1)$ implies the battery level of the node at t_1

$E_b(t_2)$ implies the battery level of the node at t_2

The value of P_c is obtained based on the following equation:

$$P_c = O\ (\frac{P_t}{d^\alpha}) \qquad - \qquad eq3$$

where,

α is a constant with a value between the range 2 to 4

P_t implies transmitted power

d implies distance

P_c implies the power consumption

While the energy consumed by each node is calculated as follows:

$$E_{con} = \frac{V_{in}}{R}\int_{t_0}^{t_1} V_r(t)\ dt \qquad - \qquad eq4$$

where,

$V_r(t)$ implies voltage across test resistance

V_{in} implies input voltage

The energy consumed by the network is estimate using a linear equation such as:

$$m * field\_size + b \qquad - \qquad eq5$$

where $m\ and\ b$ are linear coefficients. Where two nodes have equal E_{avail}, that is the same communication capacity, then their buffer size availability (measured in packets/sec), is used to determine the referred node.

Other behaviour of AEDT include the re-assignment of a parent node after a given duration. This is synonymous to the concept of a round, as mentioned earlier. However, for AEDT, the duration of a round, *t* in seconds, does not take cognizance of the status of the network, which could be in the middle of a transmission process. The parent node is always awake, while the intermediary nodes transmitting data remain awake only when receiving and transmitting data. The leaf nodes are only awake for the time they are sampling and sending data.

The objective of AEDT involves the optimisation of network lifetime and energy consumption, while it considers attributes such as average end-to-end delay, and the average packet delivery ratio. It is noteworthy to highlight the term *Objective (or Objective Goal)* as used in the above description. As mentioned earlier, all DATs usually have a primary objective while running within a selected WSN, which could include an intent to minimise energy consumption, or to maximise network lifetime. The term *Objective Goal* is thus, used in the categorisation of DATs later in the study.

DBST

The DBST DAT (*Dynamic Balanced Spanning Tree) by* (A. Avokh and Mirjalily, 2010), uses a tree topology for its logical structure. Its objectives include the minimisation of energy consumption across the network, while also attempting to balance load across network nodes. It targets especially intermediary nodes, which carry a large fraction of the data across the network. It selects a head by combining the values of the remaining energy and the node's proximity to the base station. Once the head is selected, it broadcasts a signal to the network. Based on this transmission, the nodes assess their RSSI (Received Signal Strength Indicator), to determine the amount of energy required to transmit to the base station. The outcome of this is used in the head assignment algorithm in later rounds.

DBST develops a dynamic cost function that reflects the state of the network at any time instant, by monitoring attributes such as average node distance, node density, and residual energy. It assumes that all nodes are homogenous, thus using similar packet sizes, that the radio channel is symmetric, and nodes can change their transmission power as required. These are common assumptions for homogenous DATs. DBST also defines a *Round* as starting from the point of data sampling, data aggregation across the network, to transmission of the data to the sink node. Lifetime is defined in one of three ways: when the first node dies (FND), the last node dies (LND), and a percent of nodes die (PND).

2.5.3 Chain Topology-based Techniques

Chain-based DATs build a logical chain-structure across the network nodes. The nodes at the ends of the chain are referred to as end nodes, while the final node which communicates with the base station, is referred to as the chain head. Chain topologies attempt to minimize far distance transmissions by reducing the distance between communicating nodes. It facilitates equal distribution of energy consumption across the network, while each node can only communicate with its immediate neighbour node. (Sasirekha and Swamynathan, 2015).

PEGASIS

PEGASIS (Power-Efficient Gathering in Sensor Information Systems) is a DAT proposed by Lindsey and Raghavendra (2002). It uses a chain-based topology and builds its network using a method referred to as a greedy algorithm. A greedy algorithm involves a technique where each node selects the closest adjacent node as its next communication node. By using a network-based random process, each node takes turns assuming the position of the chain head. It takes the remaining energy of the node into account, and this enhances its energy minisation goals.

The radio energy model of the WSN, on which PEGASIS relies, is like that on which other DATs rely. It is discussed further below and is discussed in more detail in Heinzelman et al. (2000) and Panchal and Singh (2018).

The typical sensor radio dissipates a certain amount of energy when receiving and transmitting a bit of data. In the case of PEGASIS (as well as LEACH), this is defined by the following set of formulas:

$$E_{elec} = 50nJ/bit$$

where E_{elec} implies the energy consumed per bit transmission. A communication consisting of the transmission of 50 bytes would consume approximately 2.5mJ (2.5 milli-Joules). The following represents the formular for the transmitter amplifier:

$$\in_{amp}= 100pJ/bit/m^2$$

where $\in_{amp}$ is equivalent to one hundred pico-Joules, per bit, per metre squared and represents the energy consumed per bit per meter squared of radio transmission.

Nodes are assumed to have control over the power of their radio transmitters and can adjust this value based on the distance to the destination node. There is an expected loss in channel transmission represented by r^2. The following equations are used to calculate the transmission and reception costs for each *k-bit* message over a distance *d*. For transmission, the following equation applies:

$$E_{Tx}(k,d) = E_{Tx} - elec(k) + E_{Tx-amp}(k,d)$$
$$E_{Tx}(k,d) = E_{elec} * k + \in_{amp} * k * d^2$$

while for reception, the following equation applies:

$$E_{Rx}(k) = E_{Rx-elec}(k)$$
$$E_{Rx}(k) = E_{elec} * k$$

where k represents packet length (2000 bytes for PEGASIS), d is distance (d^2 = 500 for PEGASIS), and the cost to transmit a packet is twice that needed to receive due to energy being split between the amplifier and the transmitter electronics. Just like most other DATs, PEGASIS assumes a symmetrical channel, which implies that transmission in both directions have the same impact on resources.

The following algorithm describes how the behaviour of the PEGASIS DAT while in operation in the WSN:

```
    i.   Start
   ii.   nodes are deployed into the target area
  iii.   the base station queries all nodes for individual energy
         levels and location
   iv.   the base station applies a random function to select a head
         node a according to the following rules:
    v.   if this is the first round
   vi.       select a node in random as the first node,
  vii.   else,
 viii.       find the node with the largest remaining energy
   ix.   transmit to all nodes the id of the head node a, and the
         location of all nodes
    x.   the head node a, sends out a signal to determine the closest
         adjacent node b, using the RSSI (Received Signal Strength
         Indicator) signal
   xi.   head node a sends a signal to node b to inform it of its role
         as next node to select neighbours
  xii.   node a sets node_role = chain_head
 xiii.   node_count = total_no_of_nodes
  xiv.   repeat while node_count > 0
   xv.       node b marks the communicating node as its next_node
  xvi.       node b sends out a signal to detect the closes adjacent
             node c
 xvii.       if node c is discovered
xviii.            node b marks node c as its pre_node
  xix.            node b sets node_role = intermediary_node
   xx.            node b sends a signal to node c informing it as the
                  next node to determine its neighbouring nodes
  xxi.       else
 xxii.            node b sets node_role = end_node
xxiii.   node_count = node_count - 1
```

```
signal_received = false
await signal to sense from base station
while signal_received = true:
    for each node:
         if node_role = end_node:
              transmit to next_node
for each node:
    signal_received_from_pre_node = false
    if signal_received_from_pre_node = true
         aggregate data with incoming data
         transmit data to next_node
         if node_role = chain_head:
              transmit to base station
End
```

The above algorithm indicates that PEGASIS operates by developing a chain starting from the sink node up to the farthest node. When each node receives data from a preceding node, it aggregates the data with its own and transmit it to the next node. When the data arrives at the head node, the head node it sends it to the base station.. In the above algorithm, chain formation is built with the help of the base station. PEGASIS requires location awareness, that is nodes need to be aware of the location of other nodes. Thus, the base station provides this information.

By choosing only nodes in proximity, PEGASIS is ains to achieve its goal of minimising energy consumption. The approach used to build the chain, as described in the above algorithm, is referred to as a greedy algorithm.

Variants of PEGASIS include include: Distributed PEGASIS (DPEGASIS) (Kulshrestha and Mishra, 2017), Modified-PEGASIS (Gupta and Saraswat, 2014), Multi-Chain PEGASIS (Jafri et al., 2013), MH-PEGASIS (Chen et al., 2011), and many others. Each one has applied a modification to the PEGASIS algorithm to attempt to improve its performance.

Most technique variants indicate attempts to improve specific weaknesses of primary techniques. In a similar scenario, research attempts have been made to combine two or more techniques to leverage the efficiencies of the individual techniques. Such attempts propose various combinations of such techniques as hybrids to solve problems encountered with individual techniques (Sasirekha and Swamynathan, 2017, Chen et al., 2011). Such techniques are discussed further in section 2.5.5 where the objective of developing them is compared with that of a dynamic selection approach.

2.5.4 Flat Topology-based Techniques

For completeness, techniques based on a flat topology are also discussed in this section to highlight their differences when compared to hierarchical topology-based techniques. Techniques using flat topologies do not make use of a hierarchical communication strategies. Each sensor node in the network plays an equal role during topology development, sensing, data aggregation, and transmission to the base station (Mamun, 2012). Techniques based on flat topologies usually use broadcasts as a data communication mechanism, while not considering the energy level of the nodes. Common problems encountered in such topologies include implosion (duplicate packets being circulated across network with nodes continuously receiving same packets) and overlap (two nodes in proximity, sensing and transmitting at same time to other nodes). Benefits include the development of good quality routes from source to sink and minimal topology maintenance. Some examples of such techniques include Directed Diffusion and SPIN. These are discussed further below:

Directed Diffusion is built on top of a mesh, or flat, topology. It includes concepts such as an *interest message*, which encapsulates the query sent into the network, and contains a set of requirements for a sensing task.

Adaptive Directed Diffusion has been developed by researchers (Sayyad et al., 2010), who have highlighted that the route could be lost when the sink node changes location, leading to more energy loss. They have suggested using learning automata to solve the problem. The unbalanced energy consumption of Directed Diffusion is also targeted by (Hao Qi et al., 2011), who proposed EAADD (Energy Aware Adaptive Directed Diffusion), which uses a different approach to building reinforced paths across the network.

SPIN (Sensor Protocols for Information via Negotiation) is based on inter-node negotiation for information dissemination within WSNs. Sensor nodes sample data and disseminate the information to other nodes in the network, making all nodes potential sink nodes. Eventually all nodes gain a view of the entire network state. Its operation is based on negotiation and resource-adaptation (Kulik et al., 2002). Thus, each node within SPIN negotiates with another node before sharing data, using meta-data to describe the data being shared. Each node has a resource manager which is consulted before any transfer of data. A node is able to evaluate the costs of sending data and proceed to send without negotiation. Identifying the potential for *blind forwarding* and *data inaccessibility* in SPIN, further research (Luwei Jing et al., 2011) has suggested a variant named the SPIN-1 algorithm, to improve its performance by extending its metric for the network's lifetime.

Hybrids involving Directed Diffusion and Cluster topology have also been developed by others (Xinhua Liu et al., 2006), with the goal of improving its energy efficiency. They proposed DDBC (Directed Diffusion Based on Clustering), which uses a passive clustering strategy, where cluster heads are selected based on a trade-off between network lifetime and energy efficiency.

2.5.5 Dynamic and Adaptive Selection of Data Aggregation Techniques

Attempts focused on developing variants of standard techniques, as well as hybrids of multiple techniques, highlights the fact that every technique or variant has certain undesirable shortcomings (Grichi et al., 2017; Kuncoro and Falahuddin, 2014). Nonetheless, such variants are still defined by the necessity to optimise target objective functions as defined by the original technique (Randhawa and Jain, 2017b). An alternate approach has been used to develop more optimised techniques however, by enabling current techniques to become adaptive based on a given attribute. This enables such techniques to adjust their behaviour by dynamically modifying such attribute based on changes within the environment. For example, considering the remaining energy of the node in the LEACH DAT. The LEACH algorithm can dynamically modify its head assignment procedure based on the energy level of network nodes, by, for instance, skipping certain nodes that could exhaust their energy if selected as heads (Panchal and Singh, 2018). It has been emphasized that in order for this approach to be truly effective, the main data aggregation approach applied across a WSN must be dynamic and adaptive, by being context aware, and able to adjust itself based on the state of the network (Gasmi et al., 2018a; Jiang et al., 2017; Wang et al., 2019). Based on this, reconfiguration within a WSN is considered to be achievable on one of three levels, which include software, hardware, and data routing (Grichi et al., 2017). In the context of the current discussion, configuration is targeted at the software level and data routing. In essence, the fundamental modification involves the topology of the nodes, which eventually dictates a redirection of data flows when changes take place within the network, and which could be used to enhance the performance characteristics of the network (Chniter et al., 2018). Others have argued that given the dynamic nature of the IoT, it is essential that WSNs be more dynamic, with a high level of adaptability. In the same vein, it has been argued that the most effective dynamic and adaptive approach to WSNs should extend to completely modifying the running DAT in near-real-time, even as the scenario is running (). This could also be extended to running one or more DATs at the same time on the WSN, in order to either optimise objective functions, or to serve more than one application at a time (). This behaviour is expected to favour various emerging applications in areas such as the IoT (Gasmi et al., 2018b), renewable energy (Abidi et al., 2017), transportation systems (Karoui et al., 2017), and smart grids (Meskina et al., 2018). The next sections discuss adaptability within various techniques. Then the differences and demerits of the techniques are highlighted. Afterwards, the topology differences are discussed. Then the chapter ends with an argument to support the dynamic and adaptive selection of DATs used within a WSN during the lifespan of a running application.

Some standard techniques were updated in other research to reflect some form of adaptability in order to improve their performance. Adaptability implies that the behaviour of the technique can be modified by adjusting one of its attributes, thereby making them adaptive. Being adaptive could simply involve awareness of node energy levels while computing routing paths. Table 2.2 presents various adaptive techniques that have such features.

Table 2-2 - Data Aggregation Techniques developed based on other primary techniques for adaptability

Name	Full Name	Source	Adaptive Characteristic	Topology	Primary Technique
EERDAT	Adaptive Energy Efficient Reliable Data Aggregation Technique	(Mathapati et al., 2012)	Cluster resized based on packet loss ratio	Cluster	Primary
PEACH	Power-efficient and adaptive clustering hierarchy	(Yi et al., 2007)	lower overhead, multi-level clustering	Cluster	LEACH
DAPTEEN	Distance Adaptive Threshold Sensitive Energy Efficient SN	(Anjali et al., 2015)	Node proximity within cluster for data similarity	Cluster	TEEN, APTEEN
CEEC	Chain routing with even energy consumption	(Shin and Suh, 2011)	Centralized control, even energy across nodes	Chain	PEGASIS
AEDT	Adaptive Energy Aware Data Aggregation Tree	(Virmani et al., 2013)	Adaptive packet reception based on buffer size	Tree	Primary
EAADD	Energy Aware Adaptive Directed Diffusion	(Hao Qi et al., 2011)	Adaptive algorithm for building reinforced paths	Mesh	Directed Diffusion
LADD	Learning Automata on Directed Diffusion	(Sayyad et al., 2010)	Learning Automata to detect sink node location	Mesh	Directed Diffusion

Tables 2.3 and 2.4 present the comparison of various DATs, showing the different values that can be held for a series of attributes. The different values affect the behaviour of the DAT when used within a WSN. Given that each DAT has a primary objective, it highlights the fact that a single DAT cannot be considered optimal across all scenarios. It also implies that during the lifetime of an application, which could consist of changing scenarios, a single DAT would not perform optimally throughout its lifetime.

Table 2-3 - Attribute Analysis for DATs across topologies – composed from (Ari et al., 2018; Mamun, 2012; Talele et al., 2015; Zanjireh and Larijani, 2015)

Parameter	LEACH	HEED	AEDT	DBST	PEGASIS	Directed Diffusion
Source	*(Handy et al., 2002)*	*(Younis and Fahmy, 2004b)*	*(Virmani et al., 2013)*	(A. Avokh and Mirjalily, 2010)	*(Lindsey and Raghavendra, 2002)*	(Intanagonwiwat et al., 2003)
Topology	Cluster	Cluster	Tree	Tree	Chain	Mesh
Aggregation Method	Cluster Head, Multi-hop	Cluster Head, Multi-hop	Intermediary Nodes, root node	Intermediary Nodes, root node	End nodes, chain head node	In-Network/Node
Next node selection method	Cluster Head	Nearest Cluster Head	MEST energy required used to select next node	OSPF path to sink used to select next node	Least transmission energy	Flooding
Head Selection / Topology Build	Probabilistic function, Num. of rounds	Remaining Energy, Num. of rounds	Least energy consuming transmission	Shortest path to sink node	Least energy consuming transmission	No heads, mesh-based, all nodes in proximity
Sensor Type	Homogenous	Homogenous	Homogenous	Homogenous	Homogenous	Heterogenous
Query Type	Sink-based	Sink-based	Sink-based	Sink-based	Sink/Node-based	Node Query-based
Algorithm(s)	Cluster, threshold	Cluster, remaining energy	Tree MEST	Tree, SPT	Chain, Greedy	2-Phase Pull Diffusion
Node Connectivity	1 node for CM, Multiple for CH	1 node for CM, Multiple for CH	1 node for leaf nodes, 2 nodes for remaining	1 node for leaf nodes, 2 nodes for intermediary and root node	1 node for end nodes, 2 nodes for intermediary and head node	Multiple, unlimited
Location Awareness	No	No	No	No	Yes	Yes

Objective Goal	Reduce energy consumption	Reduce energy consumption	Reduce energy consumption	Extend Network Lifetime	Optimise Energy Distribution/ Network Lifetime	Energy efficiency, fault tolerance

Table 2-4 - Impact on performance metric for various DATs based on scenario characteristics – composed from (Ari et al., 2018; Mamun, 2012; Talele et al., 2015; Zanjireh and Larijani, 2015)

Parameter	LEACH		HEED		AEDT		DBST		PEGASIS		Directed Diffusion	
Scenario Description	Low node count/ small field size	High node count, large field size	Low node count/ small field size	High node count, large field size	Low node count/ small field size	High node count, large field size	Low node count/ small field size	High node count, large field size	Low node count/ small field size	High node count, large field size	Low node count/ small field size	High node count, large field size
Energy Consumption Range -1 | -5	Low (-2)	High (-4)	Low (-2)	Medium (-3)	Low (-2)	Medium (-3)	Low (-2)	Medium (-3)	Low (-2)	High (-4)	Low (-2)	Very High (-5)
Energy Efficiency Range +1 | +5	High (+4)	Low (+2)	High (+4)	Medium (+3)	High (+4)	Medium (+3)	High (+4)	Low (+2)	Very High (+5)	Medium (+3)	Medium (+3)	Very Low (+1)
Delay Range -1 | -5	Very Low (-2)	High (-4)	Very Low (-3)	Medium (-3)	Low (-2)	High (-4)	Low (-2)	Medium (-3)	Very Low (-1)	Very High (-5)	Very Low (-1)	Medium (-3)
Bandwidth Utilization Range +1 | +5	Low (+2)	Medium (+3)	Low (+2)	High (+4)	Low (+2)	Medium (+3)	Low (+2)	Medium (+3)	Medium (+3)	High (+4)	Very Low (+1)	Very Low (+1)

Scalability Range +1 | +5	High (+4)	Low (+2)	Very High (+5)	Medium (+3)	Medium (+3)	Low (+2)	Medium (+3)	Low (+2)	High (+4)	Medium (+3)	Low (+1)	Low (+1)
Fault Tolerance Range +1 | +5	High (+4)	Medium (+3)	High (+4)	Low (+3)	Medium (+3)	Very Low (+2)	Medium (+3)	Very Low (+2)	Medium (+3)	Very Low (+1)	Very High (+5)	Very High (+5)
Load Balancing Range +1 | +5	Medium (+3)	Very Low (+1)	Medium (+3)	Medium (+3)	High (+4)	Medium (+3)	High (+4)	Low (+2)	Very High (+5)	Medium (+3)	Medium (+3)	Low (+2)
Complexity Range -1 | -5	Low (-2)		Low (-2)		Low (-2)		Low (-2)		Very Low (-1)		Low (-2)	
Demerits	Low latency, high scalability	Low scalability as field increases	High overhead in setup phase, topology maintenance		Low fault tolerance on failed node	High latency on larger trees	Low fault tolerance on failed node, Node energy exhaustion, node isolation		Topology maintenance overhead, low latency	Very high latency on long chains, could be replaced by variant technique	Very low latency, High energy consumption in large networks	

Table 2.5 presents the distinctions between the different topologies based on various objective functions. The values assigned to the topologies are relative to each other. Thus, since there are four topologies presented, the values range between +1 and +4 for positive results such as high energy efficiency, and -1 and -4 for negative results such as low energy efficiency, or high energy consumption. Underlined values indicate that the specific topologies could interchange values based on certain WSN characteristics.

Table 2-5 - Performance of the four topologies based on various performance metrics – composed from (Ari et al., 2018; Mamun, 2012)

Performance Metric	Cluster Topology	Tree Topology	Chain Topology	Flat Topology
Energy consumption Lowest consumption (+1)	+2	+3	+1	+4
Energy efficiency Best efficiency (+4)	+3	+2	+4	+1
Load distribution Best distribution (+4)	+2	+3	+4	+1
Redundant communication Highest redundancy (-4)	-2	-1	-3	-4
Reliability Most reliable (+4)	+3	+2	+1	+4
Scalability Most scalable (+4)	+3	+2	+1	+4
Latency Lowest Latency (-1)	-2	-3	-4	-1
Lifetime Maximum lifetime (+4)	+3	+2	+4	+1
Topology management overhead Minimum Overhead (-1)	-2	-3	-4	-1
Communication overhead Minimum Overhead (-1)	-3	-2	-1	-4 Multiple messages
Control overhead Minimum Overhead (-1)	-2	-4 To maintain Tree structure	-3	-1 No structure maintained

The data presented in table 2.4 indicates the performance of the four selected topologies, given certain scenario characteristics such as field size or node count. Various other scenario parameters could be expected to modify the above presented behaviour. The assigned values are relative across all topologies horizontally. As there are four topologies, each topology takes on a value between 1 and 4. The value is negative if the objective function is a demerit, otherwise it is positive.

2.6 Problem Identification

The discussions presented on data aggregation techniques so far have highlighted the fact that they are usually designed to perform optimally given certain wireless sensor network conditions (Ari et al., 2018; Boubiche et al., 2018). To improve their performance, variants have been developed to enable them to adapt to changes within the WSN's environment, while improving their performance. Nonetheless, WSN topologies and techniques are projected to perform optimally in specific scenarios. For instance, cluster topology is suggested to perform well in real-life emergency situations such as rescue operations and traffic monitoring, due to its low latency in small scale applications. Tree-based topologies are considered more suited to non-real-time applications targeting energy efficiency. Chain topologies are preferred for real-time applications with energy efficiency requirements, above latency (Al-kahtani and Karim, 2018; Patil and Kulkarni, 2013), while directed diffusion is better suited for distributed target tracking over a wide area (Mamun, 2012). Given these strict characteristics, the unpredictable nature of the context in which WSNs are deployed, especially in the emerging IoT, motivates the need for dynamic, self-adaptation, to enable optimal performance based on different scenarios (Rodriguez-Zurrunero et al., 2018). Robust sensing approaches are required when, for example, there are various dynamic factors such as multiple information sources (such as sensors), and rapidly changing requirements defined by both human and autonomous applications, all demanding near-real-time decision making. This need becomes more complex given the various types of heterogenous sensors, measuring multiple phenomena, and providing data in various formats, such as numerical, textual and images, using different sampling rates (such as sporadic, periodic, or asynchronous), and based off single or multiple running applications, as could be found in large disaster management scenarios (Assis et al., 2016; Zikria et al., 2019). In order to satisfy these set of criteria, a WSN would be capable of dynamically modify its active DAT (or DATs) while the application is still running (Al-Tabbakh, 2017; Daiya et al., 2016). While the discussions, thus provided so far, have aimed to support the argument for the dynamic, adaptable capacity of WSNs to define the active and appropriate data aggregation technique, given the running application or applications, the next section presents various parameters used in various research to understudy the context and behaviour of WSNs.

2.7 WSN Data Aggregation Context Parameters

Various research initiatives targeted at developing adaptive WSNs have used several attributes, which are considered essential to address the requirement put forward in the last section. As discussed earlier in the chapter, several of the attributes are related to the WSN, some to the technique, while others relate to the application. Some of these attributes are used later in this study to understand the behaviour of various DATs within WSNs (Ari et al., 2018; Chand et al., 2014; Popov and Kuzminykh, 2018):

1. Field Size
2. Active Node Count
3. Packet Size
4. Packet Count
5. Initial Node Energy (Joules)
6. Total Number of Packets Transmitted by Nodes (Transmit Bandwidth)
7. Total Number of Packets Received by Nodes (Receipt Bandwidth)
8. Radio Transmission Power (Watts)
9. Radio Reception Power (Watts)
10. Node Remaining Energy (Joules)
11. Total Network Node(s) Energy (Joules)
12. Power Consumed per Received Packet (Joules)
13. Power Consumed per Transmitted Packet (Joules)
14. Channel Bit Rate (bits per second – bps)

The values of the WSN features mentioned in the list above change over time during the lifetime of an application. In a time instant or round of a WSN application, the values held by these features could be used to determine the behaviour of the running data aggregation technique. Thus, it is expected that modifying the data aggregation technique will affect the instantaneous values of the features, and thus, subsequently impact on the performance of the network.

2.8 Summary

This chapter has discussed the rudiments of wireless sensor networks, while highlighting the justification and need for the dynamic selection of data aggregation techniques in wireless sensor networks. It was emphasised that data aggregation techniques are built for specific configurations and would perform optimally when some of those conditions are met in a WSN. Since most WSNs are deployed for individual applications, and usually into inhospitable environments, their initial configuration is static, and this makes them unusable in other applications. Their resources are managed sub-optimally and the devices are discarded after the lifespan of the application. In order to improve the value of WSNs as well as make them adaptable to new applications, there is need to make WSNs more robust to handle changing context and application requirements. This would require the inclusion of the capability to modify or adjust the active routing algorithm of the network by modifying the data aggregation technique, while the application is running in the network. The following chapters. While the next chapter discusses the methodology to achieve this objective, the following chapters will also expatiate further on this concept.

3 Methodology

3.1 Overview

The chapter discusses the methods, tools, techniques, and data required to achieve the objectives. The tasks needed to accomplish these are identified and discussed on a high level, while the empirical methods that are applied in executing the investigation are also discussed. The software components essential to achieve the outcome are identified to enable planning for the appropriate set of tools required for development. The available tools that could be used for model development, data generation and analytics are also investigated. The various steps required in data sourcing and management are also discussed, including the identification of probably data sources, and the method of analysis to be used on data for model simulation and evaluation. The approach to the evaluation of the outcomes of the investigation are also discussed to ensure that the results effectively present expected outcomes and support the aim to achieve conclusive results. A summary of the expected outcomes of these activities is presented at the end of the chapter.

3.2 Introduction

This research involves the investigation of a challenge found in wireless sensor networks, which relates the effective aggregation of voluminous sampled data. The research is based on a quantitative empirical investigation of the behaviour of data aggregation methods used within wireless sensor networks, with a goal to developing a method that dynamically selects appropriate data aggregation techniques based on changing wireless sensor network application context parameters. It will involve the identification of valid control variables values and ranges, which can be used to appropriately represent the context of the investigation, as well as to develop models that can represent the system. The results of the various experiments contribute to the development of an algorithm that can dynamically select the right data aggression technique in active wireless sensor network applications. The goal includes enabling context-aware intelligence in wireless sensor network applications that can dynamically apply appropriate data aggregation techniques that optimise the active application objectives. This implies certain objectives that need to be achieved to accomplish the aim as defined in chapter one. These are discussed briefly below, where essential tasks are mentioned.

Data Sources: the first task involves the identification and definition of data that would be required for the investigation. Candidate variables that are essential for analysing data aggregation techniques in WSNs are identified, while their data types are defined. Such variables are expected to hold important values for the investigation. This step also involves lifecycle and range specification, which become essential information during model development and data sampling and generation. Generating values for these variables will be based on identifying valid tools and techniques that can be used for producing relevant data. Such tools are identified and justification for their use mentioned. The outcome of these tasks will drive further tasks for sourcing, generating, and processing the data.

Analytical Simulations: this includes the design and execution of various experiments and simulations, which lead to the development of the preliminary models used to validate various hypotheses, as discussed in chapter one. The expected outcomes are defined, and their evaluation approaches are discussed. The various tools used to achieve these are also highlighted.

Intelligent Algorithm: the results from preliminary models developed in the simulations are studied and the highly successful models are considered for integration into the intelligent algorithm. The tasks here include the evaluation of results obtained from prior simulation and experiments, followed by the identification of success criteria and the definition of appropriate parameters for evaluation. This stage also involves the definition of appropriate benchmarks for the evaluation of the algorithm, and the review of results obtained based on research objectives.

Experimental Testing: at this stage, tasks such as selecting a methodology for the experimental evaluation of the model are discussed. The approach used to select and to apply various use cases for evaluation are discussed as well. Other discussions include how the model could be integrated into an application for user application outside testing.

3.3 Research Philosophy

This section discusses the research philosophy underlying the research. As the choices made during the research are often based on the chosen research philosophy (Johnson and Clark, 2006), this subject is expatiated here to identify the ideologies which underly the decisions made during the study. It also serves to highlight the preferred research perspectives that which the relationship between new knowledge discovered and the approach used to obtain such knowledge.

This research investigates a phenomenon, by starting with the declaration of set of hypotheses, an aim, and a set of questions. In order to address these questions, investigation is carried out with the preliminary study of literature, the identification of useful variables suitable enough to address the questions, experimentation, data generation and processing, and the development of models to certify the veracity of the phenomenon. Thus, the research follows a deductive approach, as opposed to an inductive approach, where the process commences with data collection.

The research was executed based on a positivistic philosophy. This implies that it is carried out based on underlying belief that the subject of investigation is separate from the researcher and is considered an external entity, as opposed to the realism, interpretivism or pragmaticistic views. This definition expressly defines the ontological approach used in the research.

Based on this, the research is carried out with the belief that studied phenomena can be reduced to simplest forms to carry out the investigation. The phenomenon is considered observable, while data and facts describing the phenomena can be collected and studied. It is also assumed that the collected data can be used to arrive at valid conclusions about the phenomena (Burrel and Morgan, 1982).

The research is considered scientific, and thus, maintains a realistic philosophy. In contrast to idealism, this philosophy implies that the subject of investigation is realistic and explorable through data gathering and experimentation. By applying a direct realism philosophy, the underlying assumption is that the data captured, and methods applied to process the data have a direct relationship to the results obtained.

According to Heron (1996), the choice of the research subject is subject to a reflection of the values of the researcher. The justification for topic selection for this research is very in line with the experience of the researcher. Such experience includes important subjects such as systems analysis and design, past academic research in artificial intelligence, and software development. These immensely impacted upon the selection of the qualitative analysis approach for data gathering and analysis through simulation and experimentation, including the conclusions made from experimentation. This axiological stance indicates the bedrock upon which the beliefs guiding the research are based, and that the data gathering process, the data processing, simulation, and experimentation will lead to a viable and dependable conclusion.

Based on the above description, the data collection approach will be highly structured, with the potential need to collect large samples of data, which would be subsequently processed using quantitative techniques.

The research paradigm used falls within the radical structuralist quadrant. The research approaches the investigation from a need for fundamental change in various techniques used in the studied phenomena. In contrast to the subjective approach, it applies an objectivistic perspective as the study essentially involves an investigation of objective entities.

The research approach also required the operationalisation of various concepts to enable quantitative measurements. Such measurements provided the foundation for experiments, simulation, and the generation of outcomes.

3.4 Data Sources and Data Definition

In order to develop the data required for the investigation, the form and scope of the data needed to be defined. The data type is essentially numerical and will be sourced from a simulated wireless sensor environment. The application scope of the data shall be confined to wireless sensor networks. This section shall discuss the specific methods to identify and define the necessary parameters, the data types, rules, ranges, and relationships. It also discusses the storage approach to enable further analysis on the data in order to obtain interesting patters.

3.4.1 Investigation Method

The method of investigation is based mainly on simulation. This is considered the preferred approach for various reasons. A larger fraction of literature that was studied early in the study used simulation as a preferred tool to perform their investigation and in exploring various hypotheses and theories to support their proposal. This is due to the fact that the gains from using simulation are numerous and include benefits such as minimal cost to the investigation holistically, enablement of a controlled environment, minimal impact from external unrelated elements or events, and complete control over the operation of the simulation environment, such as applying configurations to enhancement accuracy, timing and behaviour modification (Alduais et al., 2018; Antonopoulos et al., 2018, 2018; Wu and Wang, 2018). The model complexity can also be pre-determined, while simulation parameters can be stored and reused in later experiments. Development and experimental cycles can be done are faster, and there are no instances of lost or unusable sensor equipment. Simulation also facilitates the opportunity for exploration of multiple options that could enhance the outcome of the research.

3.4.2 Variable Definition

Unlike the typical physical scenario, in order to control the outcome of the simulations, various assumptions must be made concerning all attributes. Such assumptions determine the acceptable values for several attributes and parameters. In the context of this study, the term *"parameter"* can be interpreted as an intrinsic characteristic of the entity being discussed, while the term *"attribute"* is used to represent a selected, interesting characteristic, attribute or feature of the entity being discussed. Nonetheless, the two terms are used interchangeably in the following discussions. The assumptions that need to be made before simulations determine the acceptable data values and ranges used in simulation. The accepted values should represent varying levels of realism and should be appropriate to the requirements of the simulation. It is expected that the assigned values give adequate experimental accuracy, and relevance to the ongoing investigation.

In order to determine the required attributes for the investigation, a wide set of relevant literature on wireless sensor networks was studied. Chapter two has provided some detail on how the articles were used to accomplish this task. The articles studied were sourced from various online directories. Journal and article portals were used for both general and specific search. The online article directories used included the following:

- IEEE Xplore,
- SAGE Journals,
- Hindawi
- Sciencedirect/Elsevier,
- Springer,
- Researchgate,
- SciTechnol

The keywords used during search included *wireless sensor networks*, *data aggregation*, *network topology*, and *routing protocol*.

Reference Management Tools

The literature review process involved an extensive study of various related articles and journals, sought from various online repositories. The selected articles were systematically sought according to their relevance to various aspects of the study and the articles themselves maintained in cloud-service virtual folders. A series of tools were also used to facilitate and automate the process. They include the following:

1. Dropbox was used as the primary tool for document storage for articles, research documentation, Microsoft Visio files, and for storage of simulation data. This provided security for all documents, as well as making it available across different disparate systems. Google Drive and Microsoft OneDrive cloud storage tools were also partially used for the purpose of article and data storage.
2. Reference and bibliography management was performed using a reference management tool named Zotero (Zotero, 2018). This was selected against various other options such as Mendeley and RefWorks, due to its perceived ease of use, enabled by its multiple channels for article inclusion including a desktop application and the internet, its integration with Microsoft Word via a plugin, and its background automatic synchronisation of document storage. It enabled the collection of articles, automatic retrieval from document repositories, organization, citing, tagging, bibliography and sharing of articles references.

Variable Selection and Definition

A systematic approach was used to determine appropriate variables for use in the investigation. From observation of literature studied, it was discovered that most articles discussing the subject of wireless sensor networks, focusing mainly on designing or proposing a data aggregation technique or routing protocol, presented the designs using mostly common pattern. It was assumed that the incentive to apply such a pattern was inherently dictated by the nature of the system under investigation, as well as to maintain a comparable standard across similar proposals. The recognized pattern is briefly discussed in the list below:

1. Objective Goal: this is often implied in the title of the study and discussed further in the introduction.

2. Introduction or discussion of problem: current state of the problem and relevant studies is discussed.
3. Network Model: a description of scope of the study, indicating specific variables, such as field size and number of nodes are described. These will tend to affect the assumptions made.
4. Assumptions: assumptions are made, such as defining acceptable values for specific variables, given the selected network model.
5. Variable Definition: selected variables are defined to represent various characteristics of the model.
6. Algorithm: the proposed algorithm is defined based on the given variables.
7. Parameter or Attribute Specification/ Metrics: performance metrics are specified to be used to evaluate the algorithm proposed. This study does not distinguish between the two terms of parameter and attribute and uses the term "attribute" mainly.
8. Simulation: simulations are performed to evaluate the proposed algorithm
9. Evaluation: the results are evaluated, using results from comparable approaches as benchmarks, and presented using analytical and graphical approaches.

A set of articles, which use recognizable patterns like the list above are presented in table 3.1

Table 3-1 - Some articles that have used the observed pattern to propose data aggregation techniques in WSNs

Article	Pattern Used
A Novel Clustering Protocol for Wireless Sensor Networks (Darabkh et al., 2017)	Algorithms Assumptions Parameter/performance metrics Simulation
PEGASIS: Power-Efficient Gathering in Sensor Information Systems (Lindsey and Raghavendra, 2002)	Assumptions Network model Algorithms Parameter/ Attribute specification Experiments/Simulations
LEACH: Low Energy Adaptive Clustering Hierarchy with Deterministic Cluster-Head Selection (Handy et al., 2002)	Assumptions Network model Algorithms Parameter/Attribute specification Experiments/Simulations
	Assumptions

AEDT: Adaptive Energy Aware Data Aggregation Tree for Wireless Sensor Networks (Virmani et al., 2013)	Variable specification Algorithms Performance metrics Simulation

Even though the aim of this research is quite different from the common aim of the studies mentioned in table 3.1, the context of the study is similar. Thus, the variable definition approach used in this study shall be based on a similar approach used in the presented studies, as well as many others that were reviewed. This study, however, trod a more holistic approach, which involved the use of common variables across various data aggregation approaches. Table 3.2 lists the steps used in the selection and definition of variables.

Table 3-2 – Approach steps used to define variables for study

Steps	**Description**
Step 1	Collate all available wireless sensor network variables, irrespective of their type, form, or representation.
Step 2	Identify relevant stages in the process of designing, developing, and executing a wireless sensor network simulation.
Step 3	Categorise variables according to the identified stages. Variables can be duplicated at this stage
Step 4	Determine the ownership of the variables, such as the entities that hold such. For instance, entities include network, technique and the application or scenario. Variables can also be duplicated.
Step 5	Determine the types of values, and ranges, acceptable for the variables. Thus, classifications such as ordinal, discrete, etc, are used here.
Step 6	Determine the static and dynamic variables
Step 7	Define relationships between variables.
Step 8	Define the typical values for the variables, as well as their acceptable value ranges.
Step 9	Define the workflow for variable assignment and use during the process of a simulation. This should define the life cycle of a typical variable and determine its impact on the outcome of the simulation.

3.4.3 Data Storage

The data that was generated from simulation was stored using cloud-based services such as Dropbox and OneDrive. The benefit of this included that the various tasks that were not related to the entire system, such as building the intelligent model and performing an initial training, could be carried out on an optimised computer. The strategy also enabled the storage of the simulation data in raw form for further analysis, which facilitated automated testing and reduced the life-cycle period for generating results and visualisations.

The data storage strategy included the automation of folder creation directly from the simulation tool. Thus, folders were created based on a combination of various settings used in an experiment, such as the number of nodes.

This approach to folder creation and definition enabled the automated selection of data generated combined from multiple simulations before plotting in a single graph.

3.4.4 Storage Format

Various data formats were used during the study. Primarily, flat files are used to store simulation output due to the need to reduce constraints to data storage while the simulation is running. For this purpose, the data format was one of CSV and tab-delimited files. Any of the two would enable immediate visualisation of the data in graphs. Beyond the simulation, the data sharing between components of the algorithm was planned to be in JSON format. This format was chosen due to its inherent descriptive features, as well as its ease of application by other software and for human interpretation. The data is written into files that are stored using the folder structure as described in section 3.4. The output of the simulation is written directly into dynamically named files in the local file system.

No specific experiments are performed at this stage of the study. However, the identified variables and attributes will be systematically stored using a relational database, MySQL in this case, while visualization is done using multiple mind-mapping tools such as FreeMind, and iThoughtsHD. Figure 3.2 presents the flow of tasks in this section and indicates the outcome of the process.

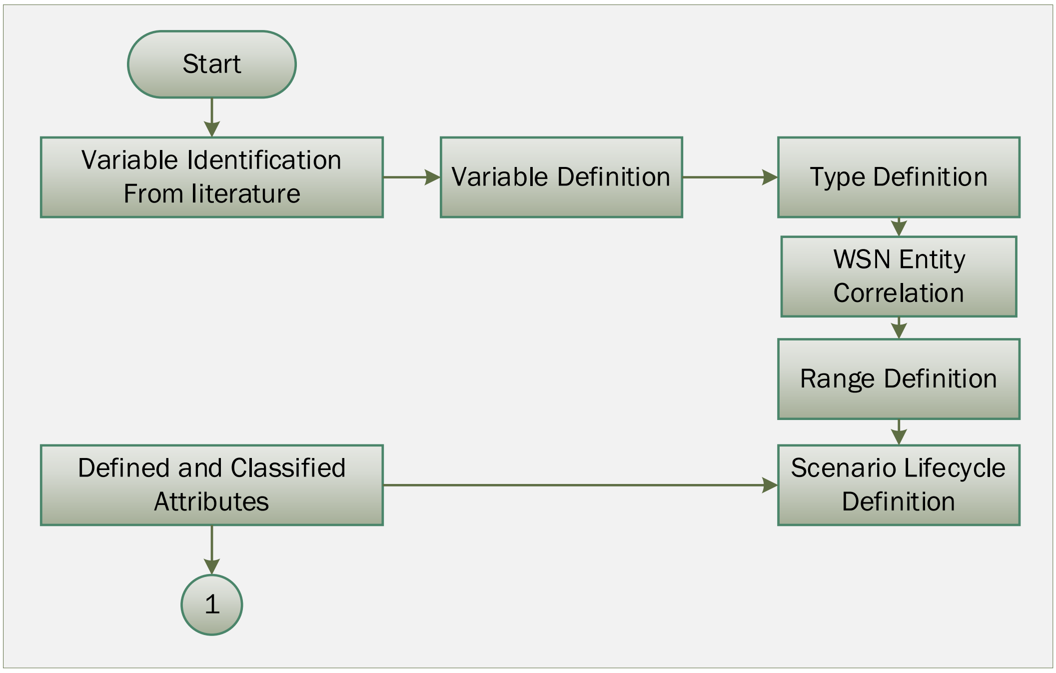

Figure 3-1 - Variable development process

3.5 Analytical Simulations

The study involves the specification of various analytical simulations necessary for validating the set of hypotheses, as well as for developing models that will be used to build the algorithm. This implies the design and execution of various experiments and simulations, leading to the development of the first sub models. This step is taken to confirm various hypotheses established in chapter one. Expected outcomes are defined, followed by the application of the selected tools to develop appropriate simulations. The various tasks are discussed in the next two sections:

Tool Selection

Various experiments involving exploratory, model development, algorithm, and prototype application development needed to be carried out during the study, requiring various tools. The available tools for each stage of the study were studied and the appropriate ones were selected based on their fit for the purpose of the study. These are discussed in the sections that follow:

Simulation

To perform simulations, the Network Simulator 3 (NS3) tool was selected. Other tools considered for this purpose included NS2, OMNET++, MATLAB, and OPNET. However, NS3 was selected based on various inherent benefits, some of which are discussed in the list below (Weingartner et al., 2009):

1. Its high relevance to research in wireless sensor networks due to the observable immense application of its predecessor (NS2) by various research articles, which were studied during the investigation.
2. It is a more recent tool and is being supported by an active academic community.
3. It is developed in a flexible high-level language, C++, and provides optional bindings for the Python language, which enables direct access to simulation data for immediate analytics. Its use of the C++ language also enables its interaction with components based outside the simulation environment, such as the operating system's folder management system.
4. It is selected above NS2 because the support for NS2 was no longer active, and the tool is based on multiple development languages (C++ and Otcl), one less common and minimising its flexibility and future relevance.
5. Using NS3 provided a strong frame of reference-based literature already studied. New studies in wireless sensors networks that relied on references that used NS2, are are more recently being performed using NS3.
6. Finally, the tool is provided with an actively updated set of documentation focusing on the goal of using the tool., accessible in PDF and in an online format.

Data Visualisation

Data generated from simulation is stored off in a folder located within the simulation environment. Visualisation of the data is essential to understand the behaviour of the simulation. Two options were considered as follows:

1. Use the in-built visualization tool within NS3, namely Gnuplot, a common library within Linux environments.
2. Python matplotlib library, which was possible based on the flexibility of NS3 to write simulation data to the appropriate directory, without further manipulation.

Both options were used in the study based on the context. The in-built visualization tool enabled immediate assessment of the simulation results, while the python libraries enabled further exploration of the data in graphs used in documentation.

Intelligent Algorithm

The intelligent algorithm consists of a related program, whose purpose is to determine, from a collection of data for each technique, and given a set of criteria, the best technique that fits a scenario. This process involves the consideration of various constraints and requirements, with respect to instantaneous time-series values of the related data, to select the best dataset that fits and suits the WSN scenario. The data is then used as feature data for the machine learning model. This component is supposed to act as a preliminary stage to the intelligent model to select accurate data for training.

Software Interface

This involves a client software that enabled accessibility to the query interface of the model. High-level, object-oriented paradigm-based languages were considered. These included Java, Python and C++. As this was considered better implemented using a web page, Python was chosen as the development language, since the Django library could then be used. The use of Python would also enable the direct integration between the model and the web interface in a future upgrade of the model.

The experiments to be carried out at this stage include the following:

- Development of models to represent different data aggregation techniques.
- Simulation to obtain technique behaviour given various environmental attribute settings.
- Evaluation of output from multiple techniques run in simulation, given similar environmental conditions

Expected outcomes include:

- Validation of need for multiple techniques in single application scenario

Figure 3.3 shows a summary of the tasks performed at this stage of the study.

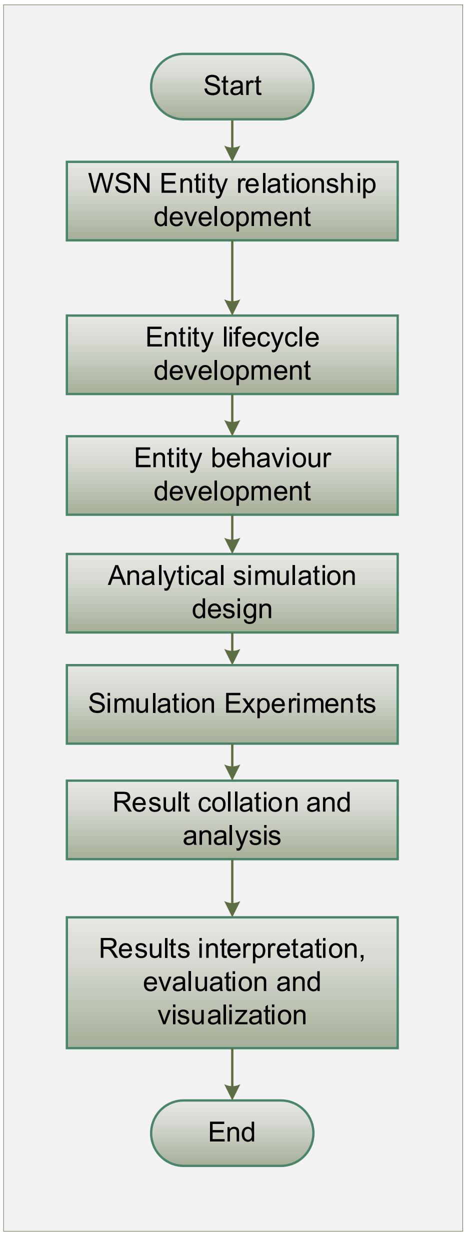

Figure 3-2 - Summary of tasks to be performed at this stage

3.6 Intelligent Algorithm Development

The approach used to design and develop the algorithm is based on a combination of various approaches taken in earlier stages and is thus, driven by the outcome of those stages. The process for designing the algorithm was guided by preliminary identification and clarification of relevant wireless sensor network attributes, each with a scope and lifecycle that is defined by the associated with the WSN context. The various experiments performed in earlier stages are used to develop sub-models with predictable outcomes and these are used to generate data that can be used in future development. Development is performed using a stepwise approach, where units of developed sub-models are evaluated and integrated into larger units. The results that present highly positive outcomes are afterwards selected to form components of the algorithm.

The variables that are required for the algorithm development were briefly identified in the literature review but will need to be selected and revised to select the essential ones that will go on to represent the WSN attributes. Such attributes are defined based on relevance to the expected outcome of the simulations.

Based on the variable definition and specification done in earlier stages, the algorithm design is also based on the outcome of correlations developed among the identified variables, including their range and type specification. This becomes important during the prototype testing stage. Variables are classified based on related WSN entities and these inherently determine the impact level of the variable to the development of the algorithm. All selected variables are defined in a similar way, and are given acceptable value ranges, data types (such as ordinal, ratio, categorical, etc.), ownerships within the wireless sensor network, and active lifecycle behaviour in a generic WSN scenario. Beyond these, various functions and relationship formulas are derived from variable relationships and will fit into design of the algorithm. These also fit into validation functions that will be used to develop the benchmarks. Essentially, these tasks are guided by studies done earlier in literature and the study of real-life use cases.

The criteria for defining successful outcomes shall be developed in order to establish benchmarks that can be used to evaluate the algorithm. The benchmarks will enable correlation with the study objectives in order to assess the effectiveness of the algorithm with respect to these.

Preliminary experiments require using the rudiments of object-oriented design and programming to encapsulate various elements that can be combined into a model of the various components within the area of study. These models will fit into the design and development of sub-models, which will be used in the hypothesis validation stage. Afterwards, these models are combined, also guided by OOP, into larger models that provide higher level functions. The corresponding benchmarks are developed for this and used to validate the models.

As mentioned earlier, experiments are carried out in the Network Simulator 3 tool (NS3), based on the C++ high-level language. The Python-binding ability of NS3 is also explored in order to integrate the simulation directly with data analytics, which can be performed immediately after each simulation run. The tools to achieve this include Eclipse and Visual Studio Code as the integrated development environments, while the environment consists of the Ubuntu operating system, running in the Oracle VirtualBox virtual machine.

The algorithm development is done in a high-level language, Python being used in this study, while PyCharm is used as the integrated development environment. The algorithm is developed in components, each representing sub-models, and tested extensively, before being integrated into a single unit for training.

The following experiments and outcomes were planned for this stage:

- Validation of results of models developed in the analytical simulations stage by using benchmarks developed for this purpose.
- The integration of various highly successful models and the subsequent simulations and evaluation of their results.

System integration and testing of the intelligent algorithm based on the various unit models. Evaluation of the system using benchmarks developed based on the objectives of the research.

Figure 3.4 shows a summary of the tasks performed at this stage of the study.

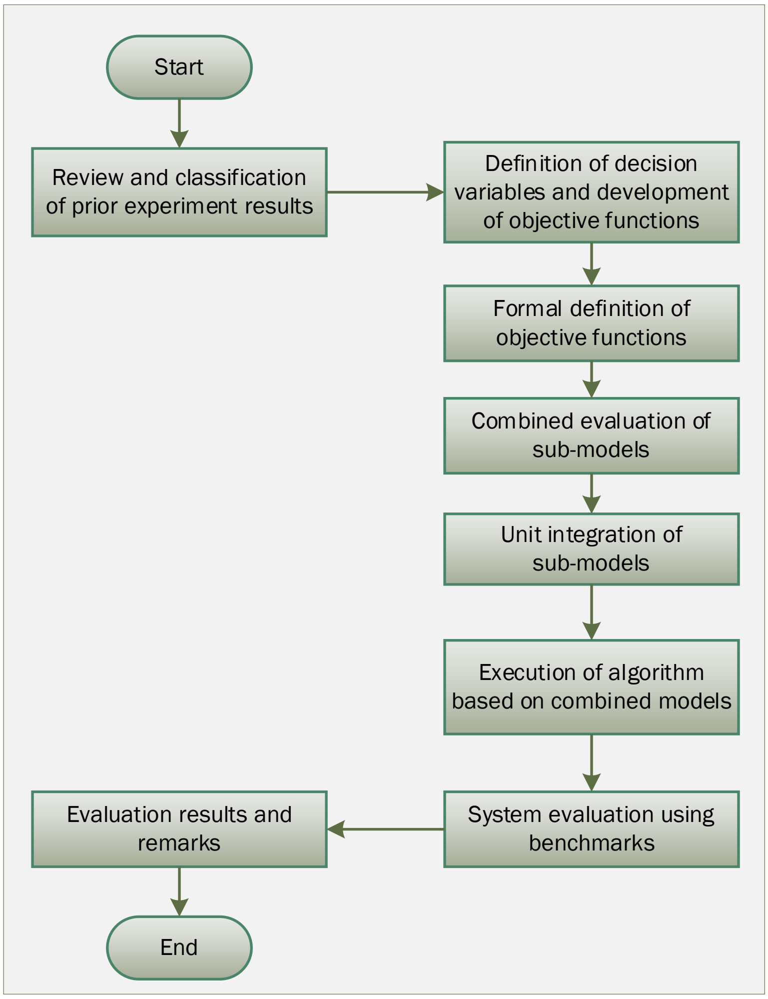

Figure 3-3 - Tasks performed under the algorithm development

3.7 Experimental Testing

The working prototype as defined in this research consists of placing the intelligent model in an application mode, which enabled testing by using various scenarios and use cases. Thus, the intelligent model would be used directly in evaluation by applying batches of use cases for evaluation. A user interface (UI) was designed in this study but however, not implemented. The use of the model directly provided opportunity for batch use case testing, as well as single scenario testing, which would have occurred via a user interface. The design for the user interface was to facilitate a first-hand, user-friendly evaluation of the intelligent model. It includes a user interface for data entry and selection, and controls for data input and execution of the algorithm. In an application mode, the intelligent model can be queried from an external source by another application. Thus, the UI-based application would have integrated with this for the same reason of obtaining a recommendation.

The execution of the working prototype to evaluate the intelligent algorithm is done using a combination of various benchmarks. Standard benchmarks are used for the evaluation of the model. Benchmarks defined in literature and obtainable from real-life scenarios are also considered as reference benchmarks. The evaluation carried out in earlier experimental stages were also collated and reviewed in order to develop new benchmarks for the algorithm. This also involved a review of success criteria and evaluation parameters within the context of the intelligent model.

Figure 3.5 presents the plan for the development of the application mode for the model.

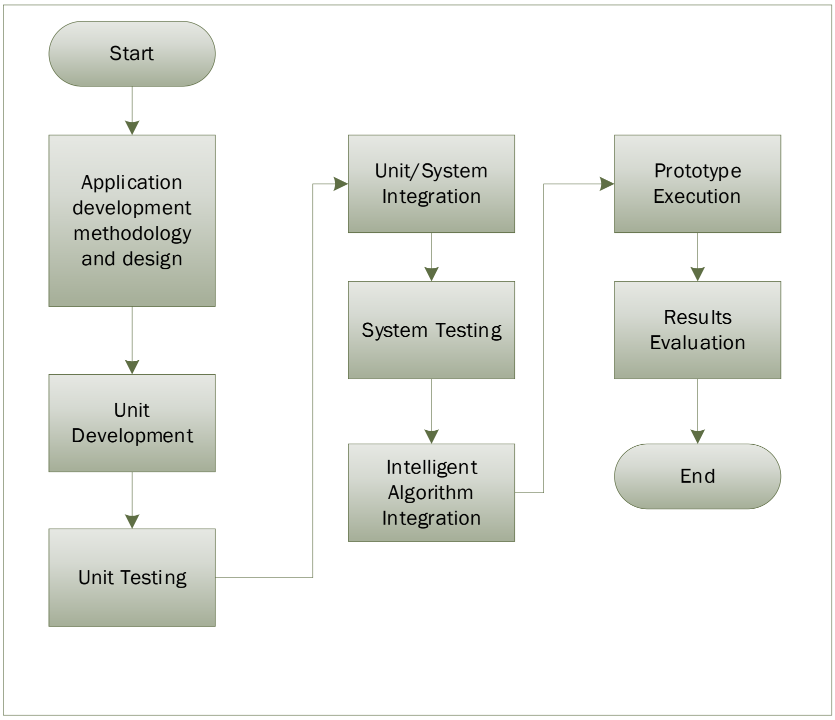

Figure 3-4 – Process used for building a working prototype for the intelligent model

The following activities and outcomes were planned for this stage:

- Selection of a design methodology, the design and development of an application that will effectively consume the services of the intelligent algorithm.
- Running of the application based on various selected use cases, including both simulated and real-life scenarios.
- Evaluation of the results obtained from testing of the intelligent algorithm.

3.8 Summary

This chapter has discussed the methodology of the research, which dictates the procedure for carrying out the research. The variable specification approach was discussed, including how data types will be defined and ranges determined. Since defined within the context of wireless sensor networks, variables are assigned to specific wireless sensor network entities, which own such data and subsequently define their lifecycle behaviour during WSN simulations. The approach to developing specific WSN models is then discussed, including methods to simulate the entities and gather data for further analysis. This leads to the development of preliminary models whose behaviour are defined by selected decision variables. These are then used to develop sub-models for the intelligent algorithm. On aggregating several models to form the intelligent algorithm, earlier collected simulation data are used in training. The results are obtained and then evaluated based on developed benchmarks. Finally, a working software prototype is developed to consume the services of the algorithm, and to provide a holistic evaluation of the algorithm.

4 Model Design and Needs Analysis

4.1 Overview

This chapter discusses the design of the intelligent model as well as presenting the needs analysis of the study. The needs analysis consists of the process of analysing and identifying the main components needed to accomplish the objectives of the study, and this is accompanied by the design of how these components will be combined to achieve the aim of the study. The chapter investigates various concepts related to the subject matter, as well as relationships between concepts and related WSN entities. It also studies and presents a strategy for the development of data aggregation techniques from a few important dimensions. It explores the relationships between these dimensions and presents a guiding process for using the dimensions to classify techniques and further understand their operation within WSNs. This process, otherwise referred to as a *Relationship Workflow Model*, is applied later in the study to guide the execution of various experiments. The chapter also covers further discussion on important theoretical entities involved in this study, related WSN scenario attributes, and how these relate to understanding the selected techniques. The chapter ends with a behavioural analysis of techniques in terms of their application in complex WSN scenarios.

The following sections explore various patterns in the behaviour of data aggregation techniques and use the subsequent understanding as a basis to highlight the need to visualise such techniques from different dimensions. The selected dimensions are then used to identify essential attributes which fit into such dimensions and are considered essential for classifying and comparing data aggregation techniques.

Three major WSN entities that are considered relevant for studying WSN application use cases are also identified and their relationships with, and impact on, WSN attributes are discussed. This enabled the development of an attribute "lifecycle", which helped define value thresholds and resolution. The developed relationship workflow is then discussed and used as a support architecture for further technique behavioural analysis. Beyond this specific use case, it was proposed that the workflow is suitable to be used as a generic guide towards the selection of appropriate data aggregation methods in complex WSN scenarios.

4.2 Preliminary Studies

Studies, that focused on developing data aggregation techniques for WSNs, have proposed new routing methods that build upon old ones in order to improve the performance of various network metrics. During the literature review performed in chapter 2, common patterns were recognised in the approach used to create new data aggregation techniques for WSNs. The initial step involved the selection of an objective function, such as minimising energy consumption or increasing accuracy, which becomes the goal of the technique. Other steps included identifying the default WSN working model for the technique, its specific algorithm(s), attribute definitions, and its method for evaluation. The evaluation of such techniques often relied on a base reference consisting of earlier well-known data aggregation techniques. Literature supporting these ideas, and the identified phases are discussed in the next section.

4.2.1 WSN Dimensions

Recall that the goal of data aggregation techniques in WSNs involves the optimisation of network metrics to improve the performance of the network. Data aggregation techniques are designed to target one or more metrics in a WSN scenario. For this reason, a technique requires a "perfect" WSN setting to perform optimally. In realistic WSN scenarios, the network model, the set of assumptions and the constraints, could change over the course of the running application. This impacts on the conditions in which the WSN operates, which could be different from the initial conditions at he start of the application. Thus, there exists an open opportunity to improve the performance of the network by dynamically changing the running technique in the WSN. By applying a systematic approach, the most appropriate technique can be determined and recommended to improve the WSN scenario's working conditions.

This section, and others in this chapter, introduce an innovative metho, d which can be used to achieve this objective. It includes the definition of a network model, a set of algorithms, a set of assumptions and constraints , and a set of metrics used forevaluation. By applying the proposed method, it was possible to detect various WSN scenario conditions, which hastened the recommendation of appropriate data aggregation techniques for the right WSN scenario.

4.2.1.1 WSN Dimensions - Assumptions

For WSN data aggregation techniques (DATs) to perform the task of optimising the performance of the WSN, their behaviour needs to be correlated with the state of the network while in operation. The primary task of the DAT involves providing a data routing protocol for the network, which enables it to perform effective data aggregation. A data routing protocol implies the definition of a path through a network of nodes, where data can be transmitted. The path is usually built using wireless communications between adjacent nodes. The conditions of the network and application have a direct impact on the behaviour of a DAT. This affects how well it achieves its primary objective function.

One important way for WSN networks to optimise their performance is to be context aware. This means that they can detect changes in the application environment. A few constraints, which affect the performance of a DAT, include the working topology of a WSN. Since, topologies are important attributes of DATs, for a technique to perform effectively, its default topology (obtained from its "perfect" WSN) must align with the network structure of the active WSN. Literature on data aggregation techniques refer to requirements such as "expected network states", where this were referred to as DAT "*assumptions*". This could also be viewed as the DAT's pre-set working conditions. For instance, the initial energy level of a sensor forms an important operating constraint for a technique (Ghai and Katiyar, 2016). Sensor devices are commonly assumed to consist of a single sensor, which specifies the resolution capacities of such devices (Harb et al., 2014b). Settings such as these could be considered as boundaries to the working conditions of a WSN technique, otherwise referred to as *Assumptions* (Omosebi et al., 2018).

Assumptions can be used to define the initial network conditions for a technique to operate. A technique with an objective to reduce energy consumption in a network could use a special algorithm to achieve this objective. Such algorithm could require, for instance, that the network nodes have prior knowledge of the logical structure of the network (Huang and Zheng, 2012). In another case, sensors are also assumed to all have a fixed compression factor, generate fixed-sized data packets, and the network is expected to have a symmetric radio channel (two-way communication), to enable a technique to effectively balance energy across sensor nodes (A. Avokh and Mirjalily, 2010). Assumptions could also include the state of node mobility in the network, i.e., whether the target is moving or not. Others include sensor node location awareness, proximity to the base station, etc. Such assumptions indicate preferred working conditions for a technique when used within a WSN and could be extended as yardsticks to determine the applicability or effectiveness of a technique in such a network. In addition, such assumptions are expected to be immutable for a technique throughout the lifetime of a WSN scenario (Omosebi et al., 2018).

4.2.1.2 WSN Dimensions – Objective Function

A WSN technique, by expectation of its function, must have a defined objective function. The values of the characteristics and attributes of the technique are expected to have a direct relationship to its objective function. These also have indirect impact on the technique's behaviour, in relation to its algorithms and performance metrics. Thus, an *"Objective Function"* could be used as a dimension for classifying such data aggregation techniques. An objective function is essential for a technique and usually links directly to one of the main performance metrics for a WSN. A few examples from literature are highlighted below.

The Two-Tier Adaptive Data Aggregation technique (TTAMA) aimed to minimize energy consumption by applying a coding scheme, which indicates the selected algorithm for achieving its objective function (Riker et al., 2016, Huang and Zheng, 2012). Similarly, the objective function of minimising energy consumption by using an "efficient cluster head selection scheme" describes both the objective and the algorithm to be applied for achieving this purpose (Arshad et al., 2012). The reduction of energy consumption (the objective function), and subsequently the extension of network lifetime, described the objective to improve a WSN scenario using M2M group communication protocols for multiple scenarios (Riker et al., 2016). These and many other proposals essentially classify the technique based on the objective, and it was considered an important attribute for specifying the appropriate technique to be used in a WSN for this study.

4.2.1.3 WSN Dimensions – Specifications

To determine the constraints for a technique's operation based on the objective function, relevant variables are identified. Such variables provide a measurable frame of reference for various environment states that impact on the functioning of the technique. Within the limits of the values assigned to the technique, these variables also provide the opportunity to adjust the behaviour of the technique. Within the context of this study, they are effectively referred to as technique *"Specifications"* (Omosebi et al., 2018). Specifications could be extended to cover the entirety of attributes defined under various entities in the WSN, such as the network and the scenario.

Various research in data aggregation technique development, in the early stages, usually specify important variables at the start of defining new techniques. For instance, the number of clusters created in a network, or the network size, are often important variables to consider in the application of a technique (Harb et al., 2014a). To minimize message overhead, the 2-tier aggregation scheme TTAMA (Riker et al., 2016) maintains variables holding values for the communication settings of nodes, such as radio range and energy consumption. Other proposals include many other attributes, such as number of nodes, simulation area, simulation time, number of clusters, channel bandwidth (Rahman et al, 2016). The number of nodes on the network could be used to adjust the response of the technique with an objective function of low-cost topology construction (Beydoun et al., 2009). Since the objective function indirectly determines the important variables for a technique, the initial values for the variables are important for defining and selecting appropriate techniques.

Considering the two concepts, assumptions and specifications, assumptions are immutable (i.e., do not change in value), while specifications can be either mutable or immutable, based what they measure. Thus, "*Specifications*" indicates the variables that could be changed within the WSN context, and which provide an avenue to change the working conditions of the technique, thereby affecting its performance.

4.2.1.4 WSN Dimensions – Algorithms

The next dimension in defining techniques involves the set of algorithms. As was mentioned earlier, the techniques select specific algorithms that enable them to achieve their specific objective functions. All techniques apply one or more algorithms, while the set of algorithms could also be used as a signature for the type of technique using them. Just as the applicable algorithms used by a technique are considered essential, so also is the ability to select one or more techniques from a given pool based on the objective function in combination with a few other parameters. Examples of algorithms used in a few techniques include the following: *Shortest Path Tree* (Virmani et al., 2013) for tree construction based on the shortest path to the root, *Minimum Spanning Path* (Wang et al., 2011) for route planning that takes into consideration the energy level of the node, and *Euclidean Distance* (Mantri et al., 2013) which takes into consideration the nearest target to a node. The *sleep and wake* algorithm is popular in WSNs techniques as well, where nodes can turn themselves off while not transmitting as a strategy to reduce energy consumption (Virmani et al., 2013). In another DAT design, the computation of a cost-function in combination with the redundant data stored in sensor nodes forms part of the algorithm used to develop the adaptive nature of the DAT (Mohanty and Kabat, 2016). Based on the algorithm, the technique dynamically computes the data transmission delay for each sensor node using its position as input. Upon reviewing other techniques, it was considered that the selected *Algorithm(s)* used by a technique could be considered a reliable yardstick for selecting the right technique for use in a WSN (Omosebi et al., 2018).

4.2.1.5 WSN Dimensions – Network Model, Performance and Evaluation

Based on the above discussions, the combination of the assumptions, objective functions, specifications, and algorithms could be used to determine a technique for a target WSN. Virtually all technique proposals and implementations provided a target model best fit for the technique. In that case, it seemed reasonable that each WSN network model could be in some way correlated with a vector of the above components. Various authors proposed the expected network models for their technique in order to define the specification of the scope for its performance. These are usually presented as a set of formal mathematical equations (Riker et al., 2016; Virmani et al., 2013) or based on a textual description of the interactions among its components (Nie and Li, 2011). This strategy is demonstrated in many technique proposals.

After reviewing the above findings, it was considered essential to include an evaluation component to the proposal of a DAT. This required the selection of a set of performance metrics, expected to be associated with the objective function. Thus, a technique with an objective function focusing on minimising bandwidth utilisation would have a set of performance metrics focusing on measuring the rate of bandwidth utilisation (Randhawa and Jain, 2017a). This is an essential component in a technique design as is observable in the literature. For instance, in providing an evaluation of LEACH-C, which is an upgrade of the LEACH technique, which targets the minimisation of energy consumption, performance metrics included the measurement of energy dissipation over time, data received over time, and node lifetime (Rahman et al., 2016). Such variables essentially provided a basis for comparing the performance between the primary and the updated technique. With an objective to manage network lifetime and energy consumption, the AEDT technique used metrics such as the average end-to-end delay, average packet delivery ratio, energy consumption and network lifetime (Virmani et al., 2013). In addition, the number of communication rounds and the remaining node energy after each round, were used as metrics in the TTAMA technique, which targeted energy consumption and network lifetime (Riker et al., 2016). Other common performance metrics include aggregation time, packet generation rate (or throughput), packet count, and packet size. It should be noted that some of these metrics also fit into the classification of specifications mentioned earlier. Although, they have a direct relationship, as a specification, they provide a initial working state for the technique, while as a performance metric, they provide a means of evaluating the technique's performance. On selecting a set of metrics for a technique, the outcome of the evaluation of the technique can be compared with a benchmark defined by the performance of an equivalent technique with similar objective functions and performance metrics.

4.2.1.6 WSN Dimensions – List of Components

In summary, the identified dimensions are discussed below:

1. *Assumptions*: these are variables that are immutable for a technique throughout the lifetime of the WSN application. They also represent the immutable variables within the application context which affect the performance of the technique.
2. *Objective Function*: this identifies the WSN metric that the technique is designed to optimise. This subsequently determines various other attributes of the technique, including its performance metrics.
3. *Specifications*: these represent mutable variables, which hold initial values before the WSN application commences operation and are expected to change during the lifetime of the application. Their values at any point in time is deemed to impact on the performance of the running technique and they provide an opportunity to improve the network performance.
4. *Algorithm(s)*: this represents the special set of rules, algorithms, and procedures that the technique applies to perform its function. It also represents the primary distinguishing characteristic of a technique.
5. *Network Model*: this represents the proposed default network setup for a technique. It consists of the combination of the assumptions, objective functions, specifications, and algorithms. It forms a vector, which can be used to evaluate the performance of different techniques based on a given network.
6. *Performance Metrics*: this identifies a set of variables based on a technique's objective function, which can be used to evaluate the performance of the technique.

7. *Evaluation*: this represents the stage where variables could be used to determine an evaluation of the technique in each WSN. It is a step beyond determining performance metrics and involves applying variables to obtain a value for the performance of the technique, as well as comparing to a baseline reference, such as equivalent performance of other techniques. This stage allows determining the necessary adjustments to enhance the performance of the technique.

The dimensions shown in the list above represent different stages in the specification process of a data aggregation technique. This analysis is essential because the aim of this research requires that the different components that make of a technique will play a part in determining the right technique for a scenario. It also enables the development of a relationship between the technique, the network, and the scenario in which they operate. The dimensions are presented in figure 4.1. This figure is used as the base representation for the dimensions and shall be built upon as the discussion on the dimensions becomes more involved. Note that there are no indicated associations between the dimensions. This is because the context in which it is applied determines the structure. The relationship workflow that incorporates these dimensions is discussed in the next section.

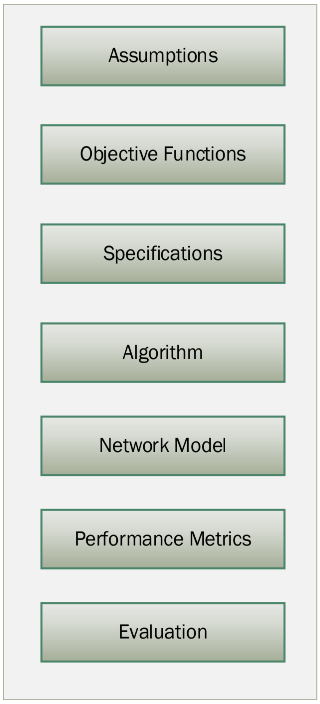

Figure 4-1 - Identification of the selected dimensions for defining a WSN data aggregation technique.

4.3 Dimensions and High-Level Relationships Workflow

Based on figure 4.1, figure 4.2 presents a more detailed diagram of the dimensions. Two different use cases have been indicated where the dimensions in figure 4.1 can be applied. The arrows are used to indicate the direction of activity. The normal flow starts from the top, and proceeds to the bottom through the dimensions. After one full loop downwards, the flow could optionally return to any of the dimensions above based on the results of the performance metrics and evaluation dimensions. Such return paths could be triggered by the integration of external factors such as a change in the context of the scenario and could require an update to the working algorithm. All dimensions can be revisited during the lifetime of the application.

The first diagram in figure 4.2, *"Process A"* indicates that all dimensions are visited until the evaluation dimension is processed. The *"Process B"* diagram highlights the possibilities during a loopback to any of the dimensions. As mentioned, a change in the context of the scenario could lead to necessary modifications to the set of assumptions, while a change in requirements could impact on the set of objective functions.

The dimensions enable a classification method for attributes of entities, which include the *technique*, the n*etwork,* and the s*cenario*. The attribute lifecycle could then be determined across three entities. The type of value, or set of values, held by an attribute could also be determined by the specific entity and dimension in which it exists.

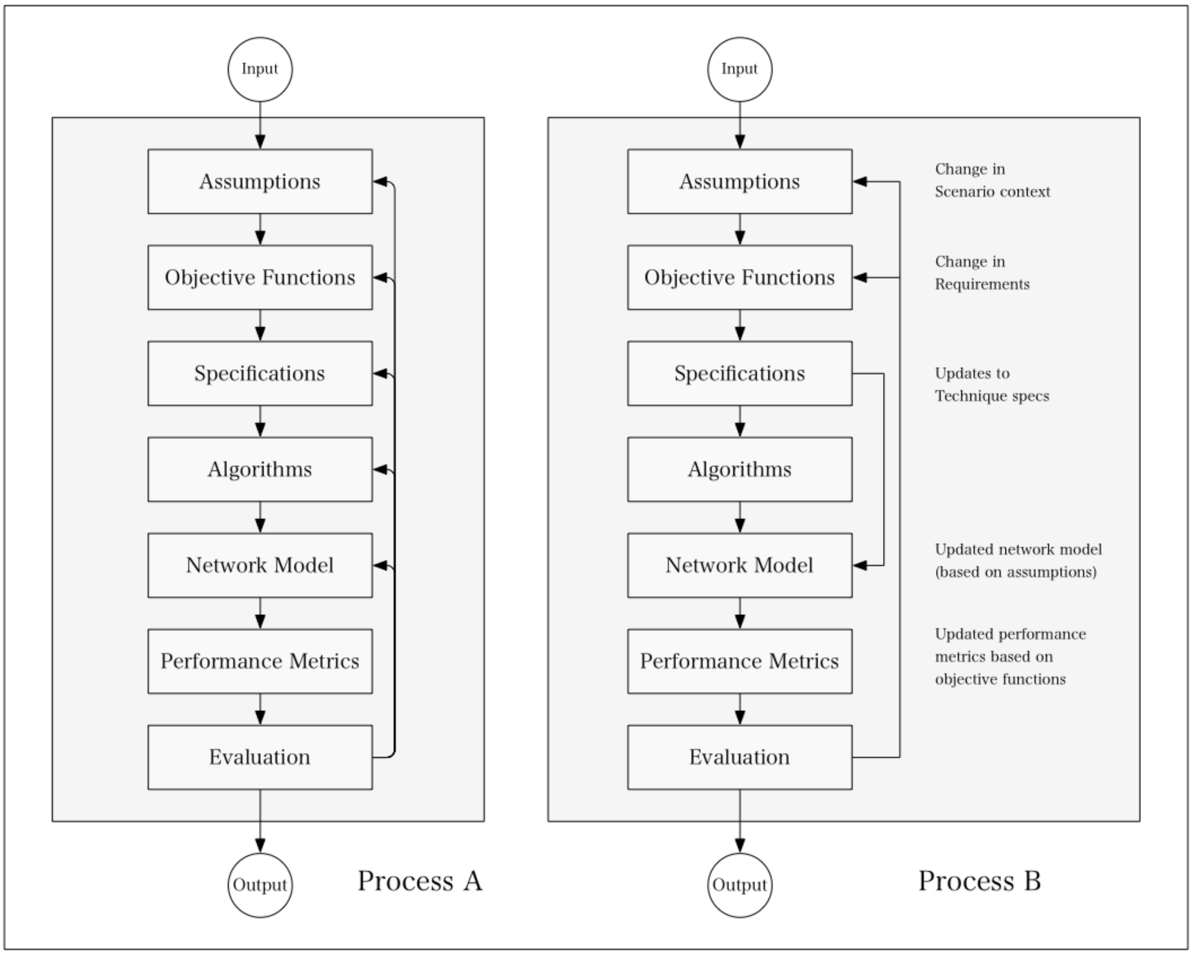

Figure 4-2 - High-level workflow incorporating the dimensions

Each data aggregation technique is defined by a set of attributes. These attributes hold values that are important for the behaviour of the technique. Based on this concept, two or more techniques can be compared and evaluated based on the value of their attributes. The task is illustrated in figure 4.3, where the attributes of two selected techniques, i.e., *LEACH* and *DIRECTED DIFFUSION* are compared across the dimensions.

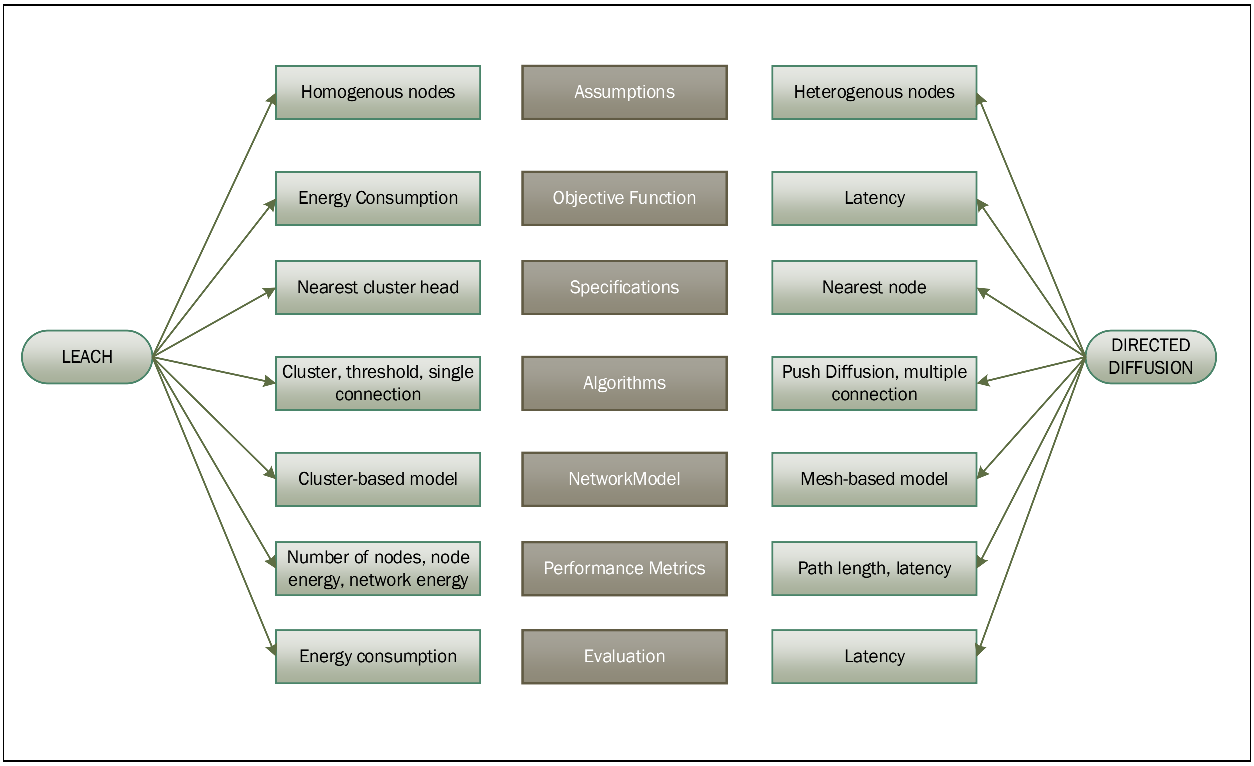

Figure 4-3 – Comparing two techniques, Leach and Directed Diffusion, based on proposed dimensions

4.4 Entities and Attribute Types

This section discusses the attributes types available for the entities, i.e., *Network, Scenario and Technique*. The *Network* entity represents a model of the WSN and includes all states and settings. The *Scenario* entity represents a model of the application, including external factors that impact on the functioning of the entire system, otherwise referred to as the application context. It represents the application event in which the WSN is being applied. The *Technique* entity models the actual running data aggregation technique. The three entities have distinct attributes which determine their working state and behaviour, impacting how they interact with each other. Their combined interaction determines the performance of the network. The relationships between the entities are discussed in more detail in section 4.4.1.

4.4.1 Entity Relationships

A *Scenario* is a collection of data and information that describe the context of the WSN application. It represents the stream of data being generated from a typical event, such as a wildfire. It could also be the source of a fixed set of data describing the behaviour of the scenario but generated from a computer simulation.

Attributes hold values that can be either static or dynamic. Static attributes do not change throughout the lifetime of the event, while dynamic attributes can change during an event and provide avenues to optimise the performance.

The *Scenario* entity represents a WSN application and its attributes. The attribute could be defined based on one of two sources: by predefined attributes (such as thresholds for energy consumption and latency, which fall under requirements), and event-detected attributes (for example *sampling rate*, which can be computed by time difference in data samples).

The *Technique* entity represents the WSN technique and its attributes. Its attributes could be static or dynamic. Dynamic attributes describe the changing characteristics of the technique during an event. Some of these have relationships with the scenario and network.

In this context, the *Network* entity represents the wireless sensor network and its attributes. The attributes hold values that represent the state of the network. Some of the attributes that fall under the network component, for instance, include the <u>*number of nodes*</u>, <u>*field size*</u>, <u>*sampling rate*</u>, etc. Changes to such attributes are expected to impact on the operation of the technique, based on its own instantaneous attribute settings.

Inclusive to the discussions presented so far, the value held by an attribute under a particular entity could be affected by one or more external factors, which are represented by attributes under the same entity or other entities. Thus, some attributes are considered primary or decision attributes, while others are dependent or independent attributes. They are discussed further below.

Given a set of conditions (determined by the scenario and network), the technique behaviour (for instance, latency, network lifetime, or energy consumption) can be determined within a margin of error. This introduces an opportunity to define relationships between the components using some mathematical formula. To improve accuracy however, the technique itself needs to be modelled, depending on the values of its various attributes, which subsequently define the technique's network behaviour and performance. Figure 4.4 illustrates how the entities interact together via the given dimensions above.

4.4.2 Attribute States and Ownership

Attributes can be associated with either a technique, network, or scenario. Their data type, value range and lifecycle are defined by these entities during an active WSN application. The attributes are defined within the context of the entities since the entity strongly determines the value of the attribute at any instance in time. Attributes have a few characteristics that determine how they are used during a WSN application, and these are presented below:

1. Static vs Dynamic: static attributes have a default value at the start of an application and remain the same throughout the lifetime of the WSN application. Dynamic attributes are those which change during the running of the application.
2. Independent vs Dependent: independent attributes are not based are not based on other attributes. They represent a specific quantity that has a direct measure of a raw measurement, such as energy consumption. They could be referred to as primary or decision attributes. On the other hand, dependent attributes are generated based on changes in independent attributes. Thus, they can be represented with functions that can be modified based on the values given to their independent attributes.

Tables 4.1a and 4.1b present associations between dimensions and static vs dynamic attributes, and dependent vs independent attributes. The inserted 'X' character implies that a dimension could contain attributes that have the checked states.

Table 4-1 - Static/Dynamic Attributes in Dimensions

Dimension	Static Attribute	Dynamic Attribute
Assumptions	X	-
Objective Function	-	X
Specifications	X	X
Algorithms	X	X
Network Model	X	X
Performance Metrics	X	X
Evaluation	X	X

Table 4-2 - Dependent/Independent Attributes in Dimensions

Dimension	Independent Attribute	Dependent Attribute
Assumptions	X	-
Objective Function	-	X
Specifications	X	X

Algorithms	X	X
Network Model	X	X
Performance Metrics	X	X
Evaluation	X	X

Figure 4.4 presents the relationship between the entities and the dimensions. It highlights the various associations between the entities and different dimensions. The figure indicates that, for instance, the technique interacts with all the different dimensions based on its attributes, while the scenario and network only interact with specific dimensions based on their behaviour during the lifetime of the WSN application.

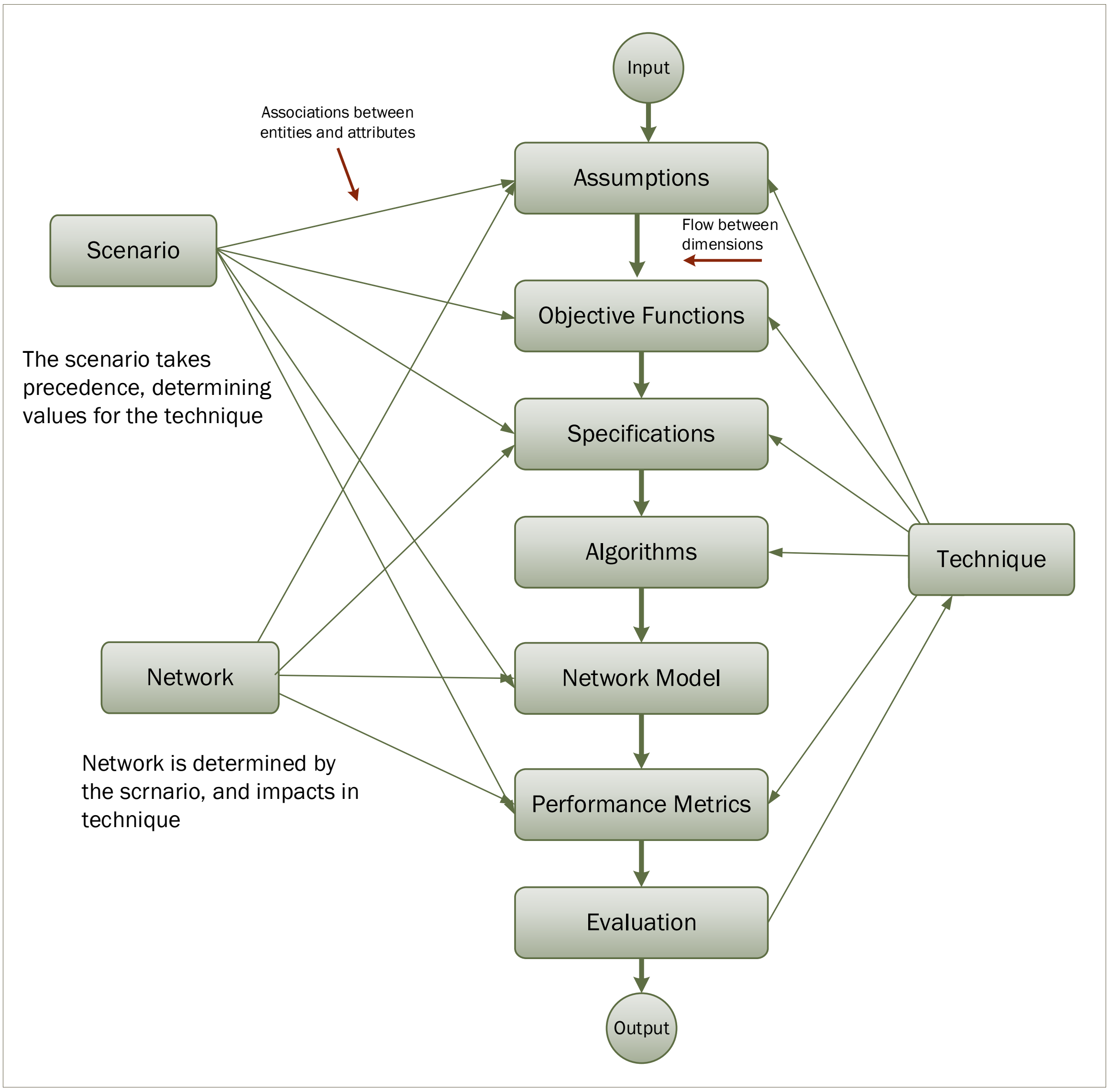

Figure 4-4 – Mid-level relationship workflow model incorporating links between the dimensions

The next section discusses in more detail the relationship workflow model based on combining the identified entities and attributes.

4.5 Proposed Relationship Workflow Model

The last section discussed the relationships between the dimensions, entities, and attributes. This section introduces the proposed relationship workflow model, which is developed based on these concepts. The interaction between the dimensions and entities, as well as attributes were highlighted in previous sections. The set of dimensions, entities and attributes can be combined to provide an accurate representation and assessment of a WSN application to enable design, execution, evaluation, and enhancement. Figure 4.5 presents the proposed relationship workflow model based on the given dimensions. The workflow model is applicable in the following scenarios. The content of the figure is discussed afterwards.

1. A dynamic WSN application, where context changes could occur, and a new technique needs to be selected based on the current context
2. A new technique needs to be selected to optimise a specific WSN metric
3. The appropriate technique needs to be selected from a set of optional techniques given a specific set of context parameters for a WSN application
4. The classification of techniques based on various assigned attributes

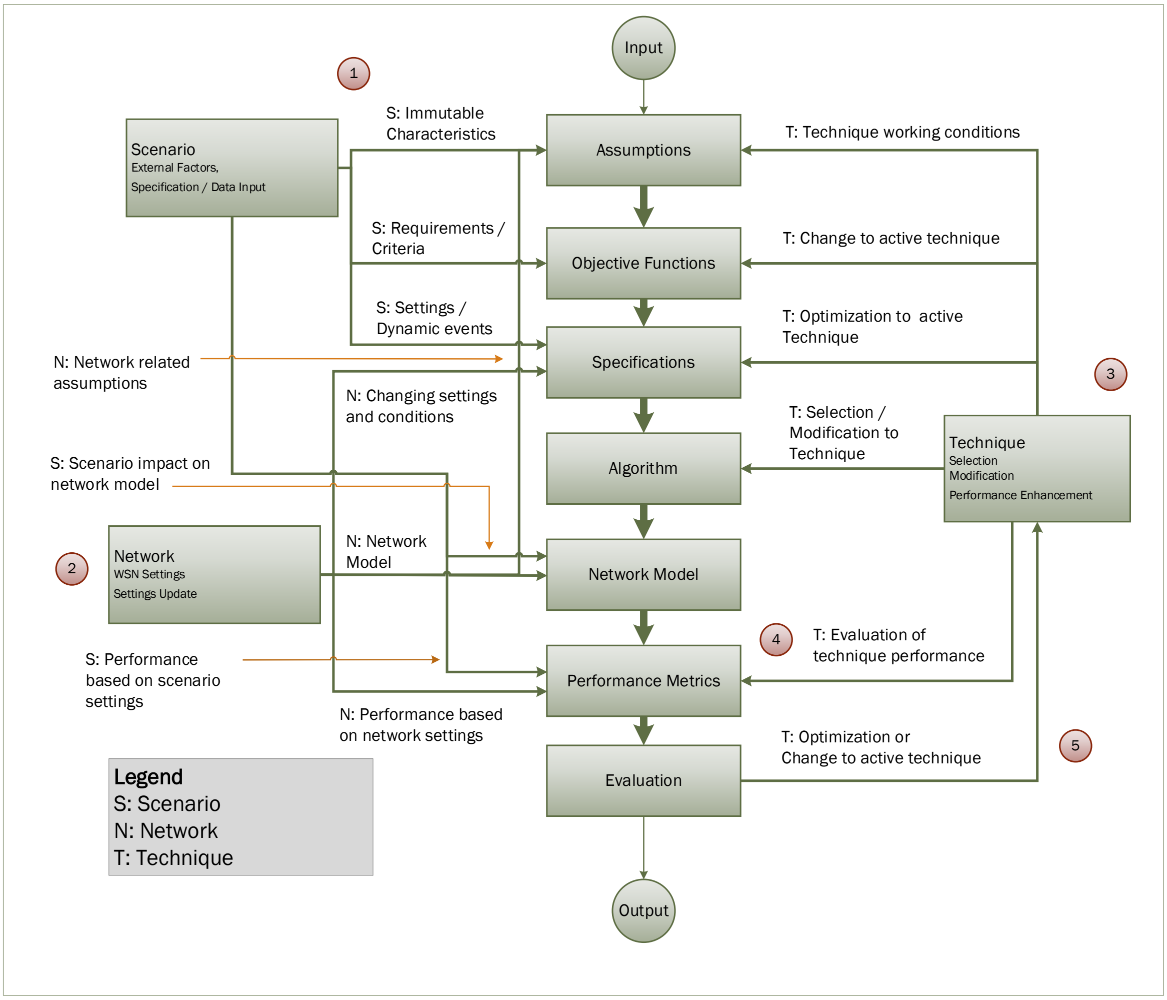

Figure 4-5 – Low-level workflow model incorporating links between dimensions

The relationship workflow model shown in figure 4.5 depicts various interactions between the entities and the dimensions. These can be combined for various workflows, one of which has been depicted by using the circles with numbers. The circled points are discussed below:

1. Scenario / External Factors: represent the scenario of a running WSN application, which could involve a simulated or real-life application. It also includes an interacting user who needs to select for the best technique-based a given WSN application. The common interaction from this box involves being able to update various attributes within different dimensions. Changes to higher stages in the dimension impact on the behaviour in stages lower down. Because this could be a real-life event, an order of interaction is not assumed. However, when the workflow is used to manage user data input, the necessary steps would start from top to bottom. A scenario would be expected to determine attributes such as the characteristics of the context, requirements, dynamic attributes, and these are expected to impact on the network model and the performance.

2. Network / WSN Settings: the network settings are usually set prior to the event. The network specifies certain assumptions, such as field size, weather, etc. Its specifications include number of nodes, field size, etc. It primarily defines the network model, and its settings have an impact on the performance as well. Several attributes are expected to change during the life cycle of the WSN application.
3. One or more of a set of techniques can be used during the running of a WSN application based on the set of requirements and criteria. Only techniques that satisfy the instantaneous conditions can be selected to be active during the WSN application's lifetime. Based on figure 4.5, the initial conditions of techniques could be taken as their set of assumptions, while their behaviour can be adjusted based on their specifications. They would also assume inherent algorithms based on their operation, while the measure of various network metrics can be used to evaluate their own performance. Based on the results of evaluation, a technique's performance can be improved by modifying its specification values, or otherwise, swapped for a better technique when the threshold of such specifications has been reached. This directly affects the working condition of the WSN application.
4. The performance metrics block indicates the attributes that are used to measure the performance of the technique. These will be used to hold values within the evaluation block, which can then be used to evaluate the performance of the technique and to assess required changes to improve the technique's performance. Within the evaluation stage, in order to improve performance of a running technique, changes need to be made to its settings. This leads to various loop-back flows where changes are continuously applied to specifications as the application runs. Beyond a pre-defined threshold, the technique needs to be modified to achieve better performance. The specific threshold could be defined by the computation of a collection of sub-thresholds. Thus, the results of the evaluation could lead to any of the following actions:
    a. Modifications to the specification of the active technique, by making changes in the specification dimension.
    b. Change to the active technique, which implies a change to the active set of algorithms.

This stage of the process requires the presence of an intelligent and dynamic algorithm, which can determine the best selection based on various context values such as objective and criteria.

These discussions have covered the relationship workflow for the entities and how it could be used in a few scenarios. The workflow is essentially applied throughout the rest of this study and shall be revisited in various stages where the discussions presented here will be realised in implementation.

4.6 WSN Models, Attribute Definitions and Entity Relationships

This section discusses the WSM model, attributes, relationships and the systematic experiments carried out. The steps involved in carrying out the experiments includes the following:

- Setup of the environment.
- Definition and selection of the appropriate set of WSN attributes to represent identified entities within the system.
- Model definition and design for the generic wireless sensor network. This includes the selection of default settings for WSN attributes to compare their performance through the use of simulation experiments.
- Model design and development for the selected techniques. This task includes a discussion on the available attributes, their association with various techniques, and the selection of default values for the selected techniques.
- Software architecture design, which determines the coordinated execution of the simulations, data storage locations, folder and file naming procedure, and the updates to wireless sensor default settings.
- Execution of the simulations based on the above details.

The next section discusses attribute definitions based on identified attributes in literature. These will be used to build the models for the entities within the system, and to define default settings for the wireless sensor network, technique, and scenario.

4.6.1 Identified Attributes

Attributes representing the properties and characteristics of various entities within the wireless sensor network environment have types and states was defined in section 4.4.2 and illustrated in table 4.1. In this section, the various attributes associated with WSN entities are introduced and their properties and associations discussed. The selected attributes enable the design of the experiments, which are later executed in this chapter. Table 4.3 presents the attributes, their characteristics, and association with mentioned entities. The attributes shown in table 4.3 can have one of various data types. The available value types include one of binary, ordinal, and continuous. Binary attributes can hold a true of false value, indicative of a toggle-type variable. Ordinal types can hold one of a set of possible values defined within the constraint of the context of the variable. Continuous value attributes hold positive real number values that can represent any scale the value of a measurement. These usually also have the characteristic of a value range.

The attributes used in this research have been selected based on literature on data aggregation techniques used within wireless sensor networks. Various techniques have used variables that are common across other techniques and valid within the context of wireless sensor networks. The discussions presented earlier about attributes applies directly to these variables. The attributes identify the characteristics of their associated entities. The entities, their related attributes, as well as the values assigned to these attributes, all represent the holistic WSN application model (*interchangeably referred to as a system*). The following are observed for interpreting the details in table 4.3. The complete table is shown in appendix A:

1. All values are based on a single lifetime of a WSN application. This period could involve the active execution of a single scenario or a complex one (multiple scenarios) and could involve one or more techniques.
2. Concepts such as dynamic and static are defined within the scope of a single WSN lifetime. For instance, a variable is considered dynamic when its value changes across two or more instances if data capture, or instance of time, occurring within the lifetime of the WSN application's lifetime.

Table 4-3 - WSN Attributes selected based on literature on wireless sensor networks

No	Attribute	Sample Values	Data Type	Primary/ Derived	Static/Dynamic	Technique	Network	Scenario	Comments
1.	Node Count (A. Avokh and Mirjalily, 2010)	50, 100 no	Continuous	Primary	Dynamic	-	X	-	Dynamic based on active nodes
2.	Topology (Mantri et al., 2013; Wang et al., 2011)	Cluster, Tree	Ordinal	Primary	Static	X	-	-	-
3.	Homogeneity (Yi et al., 2007)	Homogenous, heterogenous	Binary	Primary	Static	X	-	-	-
4.	Field Size (Beydoun et al., 2009)	100 metres	Continuous	Primary	Static, Dynamic	-	X	X	Network nodes distribution or Scenario event perimeter
5.	Network Structure (Mamun, 2012)	Hierarchical, Flat	Ordinal	Primary	Static	X	-	-	-
6.	Node Mobility (Gnanasekaran and Francis, 2014)	True, False	Binary	Primary	Static	X	X	-	Technique standard requirement or network specification change
7.	Location Awareness (Al-Karaki and Kamal, 2004)	True, False	Binary	Primary	Static	X	-	-	Node has information about its location
8.	Network Awareness (Al-Karaki and Kamal, 2004)	True, False	Binary	Primary	Static	X	-	-	Node has information about other locations of all nodes
9.	Node residual energy (Chand et al., 2014)	10 Joules	Continuous	Primary	Dynamic	X	X	-	Technique performance metric or network specification update

Table 4.3 presents selected attributes that are used later in the study. To understand the strategy for selecting the attributes, in conjunction with the entities, the following discussion is presented.

Figure 4.6 presents a class diagram showing the relationships between the three entities, attribute characteristics, and the entire WSN system. It also introduces two more components into the system. These are discussed below:

1. Performance Indicators: these measure the performance of the active technique and thus, the wireless sensor network, based on the ongoing scenario. It is determined by the attributes under the Performance Metric dimension as discussed in earlier sections.
2. PriorityList: this provides an input into the decision-making process by identifying the priority of the objective functions and other requirements that affect the evaluation of the process.

The two components provide input into the Evaluation dimension within the workflow, and thus, provide input to the intelligent model decision making process. The numbers in circles within figure 4.6 are discussed below:

1. There is only one instance of the system, represented by WSN Application. The WSN Application consists of one or more *Scenarios*, one *Network*, and one or more *Techniques*. Within the same WSN Application, two other components exist. These are the *PriorityList*, and the *PerformanceIndicator* classes. These satisfy specific functionality within the system to enable the execution of the evaluation.
2. There can be one or more instances of a Scenario within the system. The system response is based on a set of requirements, which are linked to a specific scenario. However, the inclusion of more scenarios would lead to labelled requirements used to identify each scenario. Nonetheless, the response of the system will be based on the combined assessment of the requirements and performance metrics. In this case the *PriorityList* would be required to prioritise the decision taken, which would be based on the ordering of the objective function.
3. There can only be one Network at one instance (within the context of this research) since the nodes can physically (as well as in simulation) only exist in a single network. It is, however, possible for nodes to exist on multiple logical topologies, which implies that they run more than one technique at once. However, this is not

considered in this study. It is expected that one or more scenarios could be running within the single network, and thus, the network would have to respond adequately.

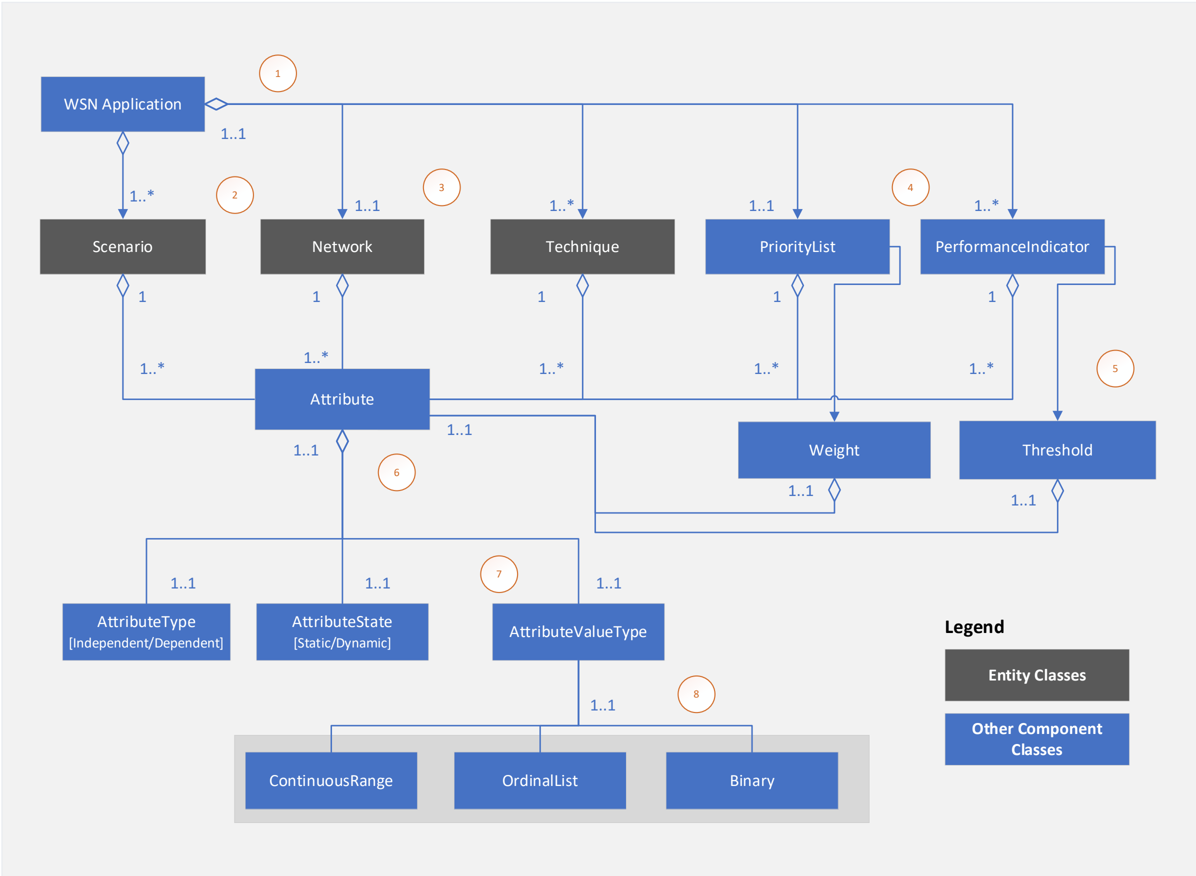

Figure 4-6 - Class diagram showing the relationships between entities, attribute and two components in the system (i.e. PriorityList and PerformanceIndicator)

4. Within the system (WSN Application), there can be only one *PriorityList*, while there exist one or more *PerformanceIndicator(s)*. The priority list provides an order of concerns, which defines the order in which the objectives are met, in the case where there is more than one objective. This list is expected to change as the system is in operation. The *PerformanceIndicator* would be linked with the *Objective Functions* dimension and should also link to the *Evaluation* dimension. This class identifies the active list of performance indices, and these are used by the *Evaluation* stage for its own task.
5. Each *PerformanceIndicator* has a threshold range. This is used to evaluate the level of performance of the network. This range can be modified by direct input or by external forces within the system. A specific attribute of the PerformanceIndicator will however, hold this threshold characteristic. Likewise, the *PriorityList* consists of a set

of *PriorityItem(s)*, which each have a weighting. This entity is not shown in the figure in order to enhance clarity. The weighting also determines which priority tops the list when the evaluation needs to be performed.

6. An attribute is defined by three components, which are *AttributeType*, *AttributeState*, and *AttributeValueType*. These are defined by the location of the attribute within the set of dimensions and its ownership among the entities. The three are considered components of the attribute in object-oriented parlance.
7. The *AttributeValueType* can have one of a set of options. These consist of *ContinuousRange*, *OrdinalList*, or *Binary*.

The class diagram presented in figure 4.6 serves as input into the development of the intelligent algorithm, which selects the right technique based on generated simulation data. This is discussed further in chapter 6.

4.6.2 WSN Model and Diagram

This section discusses the target WSN model as used for the simulation. It introduces these here in order to provide a source of reference for further discussions in later sections. The discussion focuses on the settings used in defining the simulation environment. The wireless sensor network consists of a starting number of active nodes, distributed randomly into an area measuring $50m^2$ up to $150m^2$. Each node in the network is expected to be homogenous, have the same capabilities in computation, memory, and network communications. The nodes have the capability to modify the power of their radio transmitters to reach their destination node during communications and they all have the same starting energy level. The base station is external to the wireless sensor network and located at a certain distance to the network. The default settings of the network are presented in table 4.4. These values are not static and could be modified in new simulations, as well as being modified during the process of a simulation. The values in table 4.4 are stored within the *Network* WSN entity. The values are reflective of a typical physical WSN application scenario.

Table 4-4 - Default values for the WSN model

Attribute	WSN Default Value	Conditions for change
Number of initial nodes	100 nodes	Can be modified to affect the response of the running simulation
Field size	50 m x 50 m (meters^2)	Can be modified based changing event coverage
Initial Node Energy	50J	This can change as the nodes start to communicate
Sampling rate	600kbps	Can be modified by changes in scenario characteristics, such as temperature
Base station location	X = -25m, y = -25 (based on x, y coordinates)	This should not change for each simulation since only static targets are considered
Energy consumption per bit	50 nJ/bit (nano-Joules per bit)	This remains static during the simulation
Packet size	6kb (kilobytes)	This remains static during the simulation
Network density	0.01 node/m^2	This is determined dynamically by the values of the number of active nodes and the field size.

4.7 Technique Models

This section discusses the development approach and behavioural analysis of the techniques selected for the study.

4.7.1 Technique Model Development

To illustrate the structure of various deployment approaches, figure 4.7 presents two snapshots from the simulation environment. Each snapshot shows sensor nodes represented with red dots. The diagram on the left represents random deployment of 48 nodes, consisting of 12 major clusters. The diagram on the right represents 100 nodes deployed into a grid pattern.

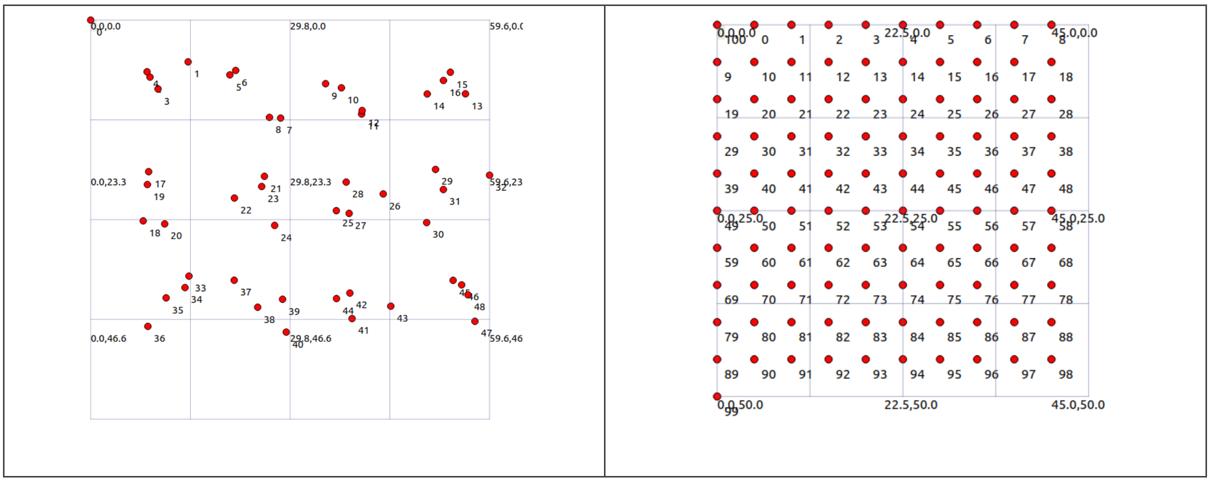

Figure 4-7 - Illustration of node deployment in the simulation environment. The left image shows 48 nodes deployed in 12 clusters, while the figure on right shows 100 nodes deployed in a grid.

The modelling of techniques in a simulation environment is necessary in order to enable controlled running of the techniques to generate essential behavioural data. The simulation environment used in this research is referred to as *Network Simulator 3 (NS3)*. NS3 is a C++ based development environment and is used to build simulations for wireless sensor networks. It also uses various terms quite like those used in this study. For instance, the concept of *Application* in NS3 refers to the running WSN application, while a *Node* refers to a sensor node in the network. The following discussions present the development strategy. Afterwards, the behavioural analysis of the techniques is discussed.

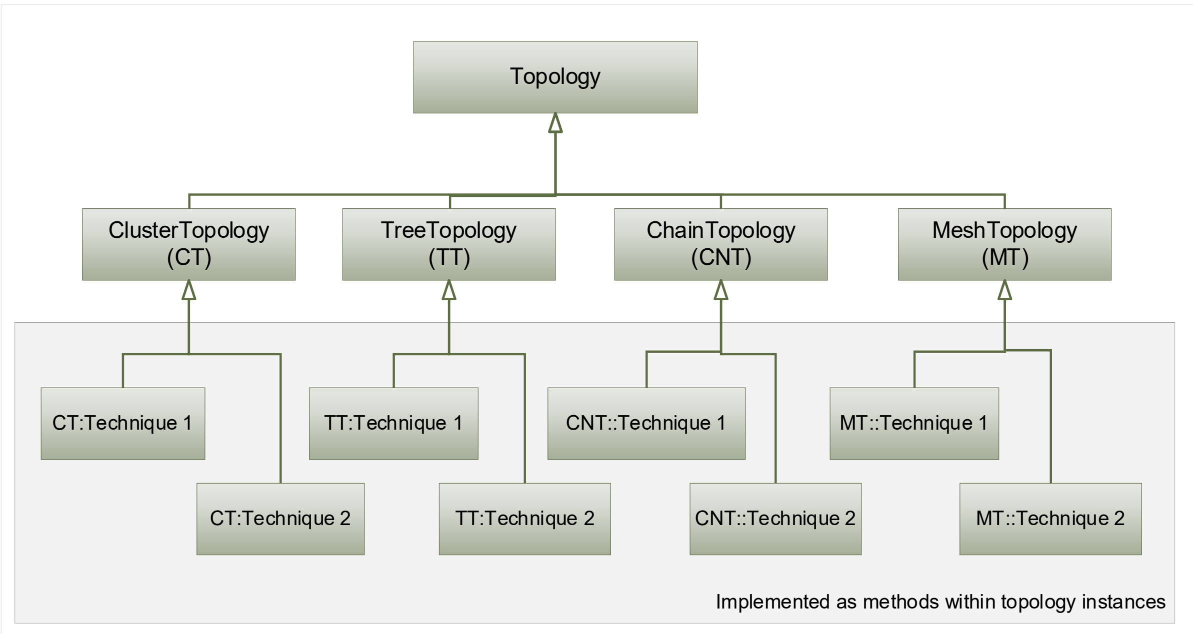

Figure 4-8 – A class diagram for the software architecture used to develop technique models in the NS3 simulator

Since the NS3 simulator is based on C++, it inherently enables development using the object-oriented paradigm. According to figure 4.8, a top class called *Topology* is created. The different types of WSN network topologies, i.e., *ClusterTopology, ChaiTopologyn, TreeTopology* and *MeshTopology*, will then derive from the parent class, Topology. Afterwards, each data aggregation technique was developed within the topology classes using specific methods. For instance, a LEACH technique was built within the Cluster Topology class using a method or function specific to the behaviour of LEACH. This strategy enabled the inclusion of new techniques by simply designing their method and inserting in the appropriate topology class. All common data aggregation technique behaviour were encapsulated in the parent classes, i.e., the *Topology* class.

4.7.2 Technique Models Implementation Design

As discussed in section 4.7.1, according to the strategy used to develop the technique models, rather than developing classes for each technique, methods were used to define their essential algorithm, and included in the concrete instance of the appropriate topology class. This strategy was considered appropriate because the algorithm used by the techniques was found to be the main differentiating factor for technique performance. Table 4.5 presents the list of selected techniques used further in this study.

Table 4-5 - Techniques discussed further in the chapter and study

Technique	Topology
LEACH	Cluster
HEED	Cluster
PEGASIS	Chain
DBST	Tree

The following discussions cover the simulation of the various techniques and their graphical comparisons. The experiments discussed in the following sections are based off evaluation experiments carried out in past research in data aggregation techniques (Meena and Manikandan, 2017).

The order of discussions is as follows:

1. The simulation and plots for the techniques LEACH, HEED, PEGASIS, and DBST are presented, and their specific behavioural characteristics discussed.

2. Comparison graphs are presented for all techniques.

4.7.3 Technique Behaviour Analysis

This section discusses each technique as introduced in past sections and provides some details about their behaviour in simulation using various graphical plots. The graphs and analysis presented here reflect the implications of various research which had focused on the evaluation of DATs based on various metrics, especially energy consumption. Thus, the graphs directly reflect the conclusions reached in those related experiments.

The term *context* as used in foregoing discussions implies a, instantaneous combined state consisting of the scenario, the network entities, and the application. Once the simulation of the technique is performed, the generated data is plotted using the simulation environment's *gnu plot* graphical tool for instant visualisation, or Python's *matploltlib* library. The next sections discuss the LEACH technique, focusing more on its algorithm and its practical application.

LEACH Technique

The Leach technique applies various algorithm steps during its operation. It has a few assumptions and some of these are mentioned here: the location of the base station could be outside the network region, all nodes can communicate directly with the base station, all nodes are homogenous and energy constrained, they do not have location awareness, the network has a symmetrical propagation channel, i.e. communication can be transmitted both ways, and cluster heads are responsible for data aggregation. As with other techniques, Leach's behaviour can be classified into a set of tasks performed in a loop referred to as a *Round*. For Leach, each round consists of two phases: the set-up and steady states. The setup phase involves head selection, advertisement, and cluster formation, while the steady-state phase involves sensing and transfer of data to the sink. The energy required to transmit a bit of information between any two nodes is the same, given the same distance apart. A cluster of nodes *n* has a head node, which compresses data for the entire cluster. The head collects *n* x *k*-bit of data from the *n* adjacent cluster nodes and compresses it to *cn* x *k*-bit data, transmitting this to the sink (otherwise referred to as the base station - *BS*). Here, *c* represents a compression coefficient, an attribute that determines how much data is sent to the sink, and thus, the energy consumed by the node to send the data. LEACH also limits the number of heads within the network to 5% of the number of active nodes. The algorithm of LEACH is briefly described below:

i. Select candidate head nodes based on a probability function
ii. Candidate head nodes advertise their status as head nodes via a broadcast
iii. Other nodes receive the broadcast and select the closest head node
iv. Each node notifies the selected head node with the intention to join the cluster
v. The head node adds the communicating node to this list of cluster members and defines a TDMA slot for it.
vi. All nodes start to sense their environment and transmit to the head node of the respective cluster.
vii. The cluster head collects all data from cluster members and aggregates the data.
viii. The head node of each cluster sends the aggregated data to the base station.

Technique Modelling

The Leach algorithm, as described in the last section, was used as a guideline to develop the based technique model. The LEACH technique was implemented under the *ClusterTopologyApplication* class. Figure 4.10 shows the main components of the algorithm that were implemented. These consist of the following:

1. Head selection: based on a randomized function
2. Topology formation: based on head advertisement, cluster-head selection, and cluster head TDMA assignment.
3. Sensing
4. Transmission: that is communication from node to node, as defined by the cluster topology.

Once the simulation commences, Leach is assigned as the next technique to run. Thus, on each round, during each of the above stages, the LEACH specific algorithm is selected to implement the simulation. Whenever nodes transmit data, their energy is depleted. The simulator uses an event mechanism to record energy changes based on when a node transmits or receives communication from another node. The network-wide energy level could also be computed at the end of a communication round by summing the residual energy of all the nodes in the network.

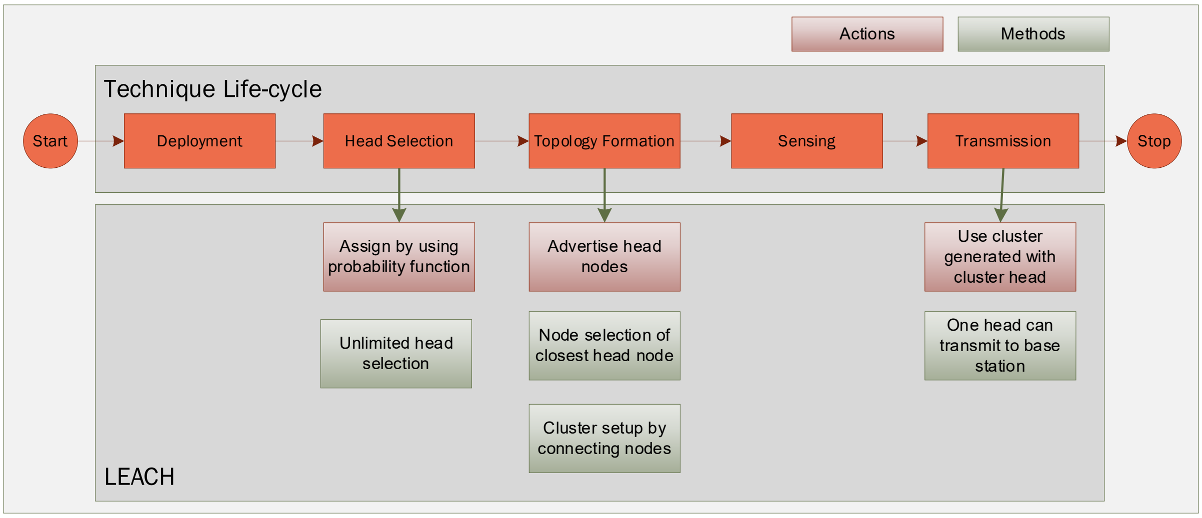

Figure 4-9 - Leach model algorithm implementation

Simulations and Results

The Leach technique is plotted in figure 4.11 based on 100 nodes across 150 rounds, over a field size of 50 m^2. The curve can be seen to start with a mildly steep curve between round 0 and about round 5, indicating that the energy consumption starts slowly. Then it becomes steeper beyond this stage and towards the top of the graph.

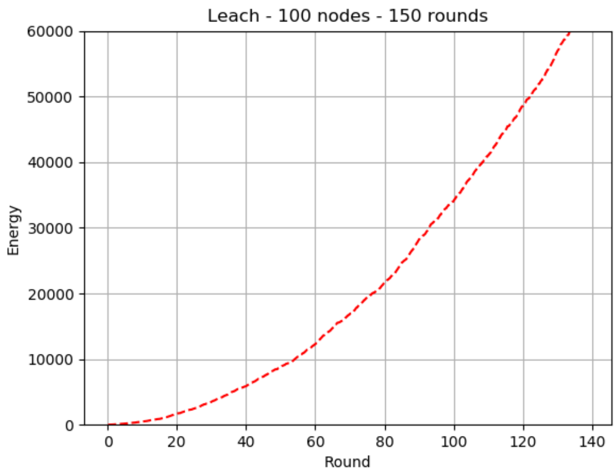

Figure 4-10 - Simulation plot of Leach DAT on Energy Consumption vs Round for 100 nodes and 150 rounds

HEED Technique

Heed is a cluster-based technique like Leach, was developed to improve on the performance of Leach by taking into consideration the remaining energy of nodes when assigned the role of cluster heads. This is combined with the probability function, which then determines if the node gets to be selected as the cluster head. Also, Heed has no constraint on the number of active heads in a round, leaving the assignment of heads to reach an arbitrary but consistent fraction of the number of active nodes. It assumes that nodes are location unaware. Nodes have a fixed level of transmission levels when transmitting to neighbour nodes. Heed does not assume any network density or diameter, energy distribution or consumption among nodes. All nodes take decisions only on locally available information. Cluster heads can also create a parallel multi-hop path to the sink node.

Technique Details

The Heed technique attempts to distribute cluster heads evenly across the network field to ensure that all nodes are connected. Its main goal is to extend lifetime of the network. A high-level description of the algorithm of Heed is presented below:

1. Perform cluster head candidate assignment based on residual energy of nodes on the network.
2. Candidate cluster heads confirm their status based on their selected probability value between 0 and 1. This determines the pre-selected constraint on the possible number of heads.
3. Cluster heads advertise their status to nodes within a defined radius.
4. Listening nodes compare distances to advertising cluster heads and select the nearest cluster head.
5. Cluster heads add the transmitting node to the list of cluster members and assigns it a TDMA transmission slot.
6. Environment sensing commences.

As discussed earlier, these steps are built into a method in the ClusterTopologyApplication class. The next sections discuss the modelling and simulation of the technique.

Technique Modelling

The software model designed for the Heed technique is shown visually in figure 4.12. Head assignment involves taking the node’s remaining energy into account, including a function based on as assigned random value for the node. In contrast to Leach, the number of acceptable heads is unlimited, and the topology formation also includes head advertisement, node selection of heads and cluster setup.

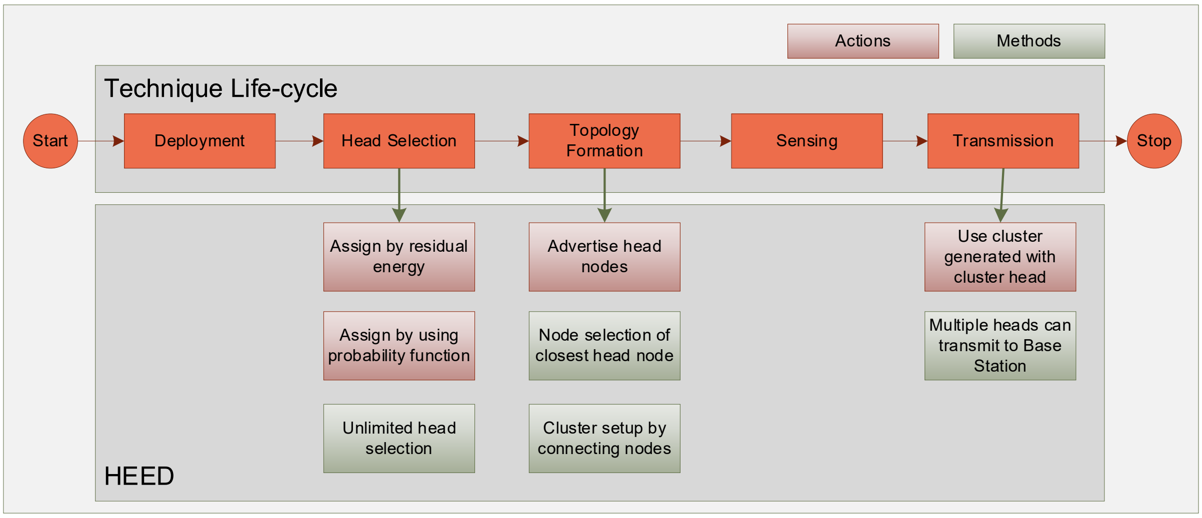

Figure 4-11 - Model design for the Heed technique

Simulation and Results

Figure 4.13 presents plots of Heed based on configurations of 100 nodes over 150 rounds. A similar curve pattern is observed here as seen in figure 4.11 for the Leach technique.

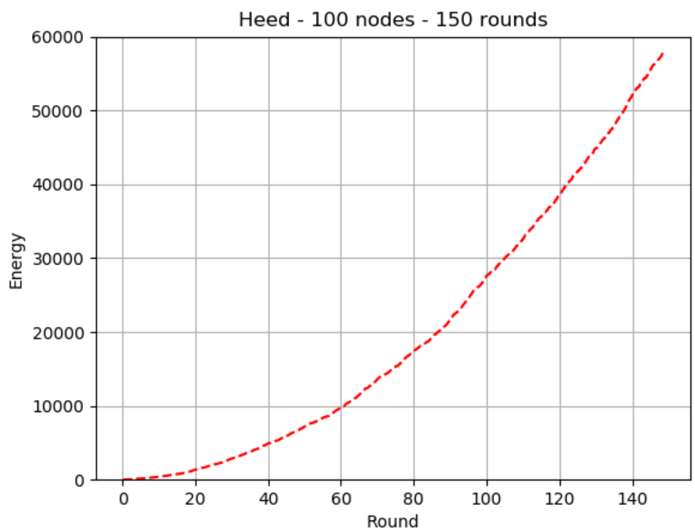

Figure 4-12 - Simulation plot of Heed DAT on Energy Consumption vs Round for 100 nodes over 150 rounds

PEGASIS Technique

This section covers the details for the Pegasis technique. It uses a chain-based topology where all nodes are aligned to form a single chain. The head can be located anywhere along the chain and other nodes need to transmit their data to the next node on the chain until it reaches the head node. Thus, transmission is based on reception from another node on the chain. Pegasis strives to achieve minimal energy consumption by minimising the distance between nodes and the number of nodes that any node needs to communicate with.

Technique Details

The Pegasis technique applies various algorithm steps to build the cluster topology. It applies various assumptions like those defined for the Leach technique. Pegasis also applies a random function in selecting a "chain head". It assumes that nodes have global knowledge of the network structure. Chain construction starts from the farthest node to the sink node and uses a greedy algorithm, where each node finds the nearest unconnected node to serve as its "*next node*". This will tend to increase the average distance between source and destination nodes as the chain grows. The chain is also able to heal by a process of node disconnection, where a dying node located between two nodes is dropped and the its two adjacent nodes are re-connected. On receiving data from its neighbour, a node aggregates the data with its own data and transmits it on to the next node. Randomness is also used as a strategy to build robustness to failure. The network communication strategy is based on a "token-passing" procedure, where the node that holds the token is the node that transmits to its next node. Lastly, Pegasis attempts to maintain even energy consumption among nodes based on its reduced distances between nodes and nodes only transmitting once during a round. The high-level Pegasis algorithm is described below:

1. Random selection of the next chain head. This will serve as the sink node.
2. Identification of the node in the network at farthest distance to the base station. This node shall be referred to as the "chain end".
3. Chain topology construction starting from the farthest node, the chain end, up to the chain head.
4. Sensing of data starting from the chain end node and transmitting up to the chain head.
5. Transmission of aggregated data from the chain head to the base station.

The following sections discuss further the modelling and simulation graphs generated for the Pegasis technique.

Technique Modelling

Figure 5.9 shows the algorithm layout for the Pegasis technique. The areas of emphasis include the head selection, topology formation, and transmission stages. This reflects how the technique was modelled for simulation. Thus, a method was developed to implement the specific functions within the *ChainTopology* class. The next sections discuss the plots of generated data from the model in simulation.

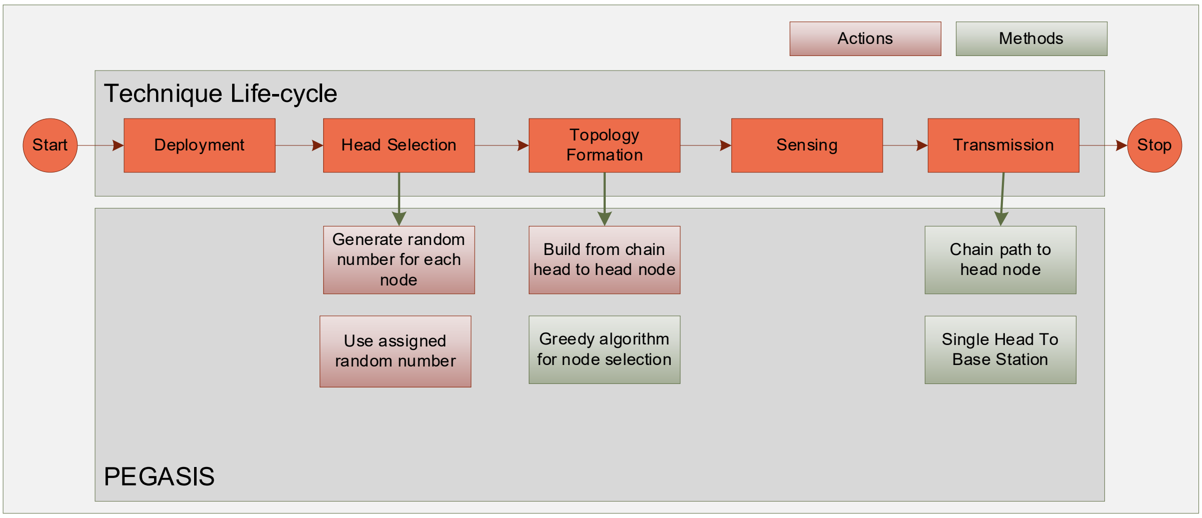

Figure 4-13 - Algorithm design for the Pegasis technique model

Simulation and Results

The simulation of the Pegasis technique was performed on 100 nodes over 150 rounds. As shown in the plot in figure 5.16, the technique presents a similar energy consumption pattern to the Leach and Heed techniques. However, its improvement is highlighted in figure 5.17, where it becomes clear that the technique has an improvement over the Leach techniques during the lifetime of the WSN simulation.

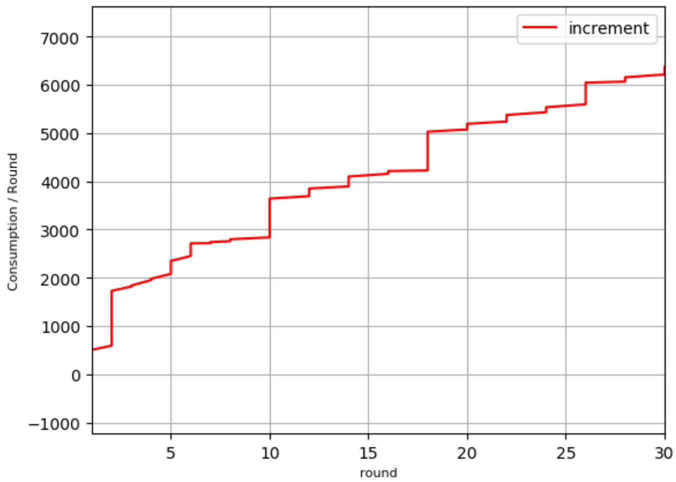

Figure 4-14 - Simulation plot of Pegasis DAT on Energy Consumption vs Round for 50 nodes over 30 rounds

DBST Technique

The Dbst (Dynamic Balanced Spanning Tree) technique selects a tree root based on a node's average proximity to the sink, followed by its residual energy. Uniquely it strives to dynamically re-arrange the tree based on the minimum energy required to sense and transmit, and energy distribution across nodes. This is aimed also at minimizing the rate of transmissions involving weak nodes in the network.

Technique Details

Under the Dbst technique, lifetime is defined in three different stages which include first node death (FND), last node death (LND), and percent node death (PND). These could also be used as a yardstick to evaluate this technique. The algorithm of Dbst is described briefly below.

- The sink node broadcasts its location to all nodes in the network using a "Hello" packet transmission
- All nodes approximate their distance to the sink node based on their Received Signal Strength Indication (RSSI)
- For the first round, the closest node to the sink is selected as the tree head. Then the hierarchical tree structure is built using a tree building algorithm.
- For rounds following the first round, the node with the highest remaining energy is selected as tree head. In the rare case where two nodes get selected, perhaps due to the resolution of the comparison, the distance to the sink would be used to select the preferred node.
- The RSSI enables all nodes to determine their distance to the sink node in conjunction with whether they have residual energy that could enable them to successfully transmit to the sink node. This factor is also used to determine the best route from each leaf node to the tree head node.

Modelling

Figure 4.16 illustrates the model design for the Dbst technique.

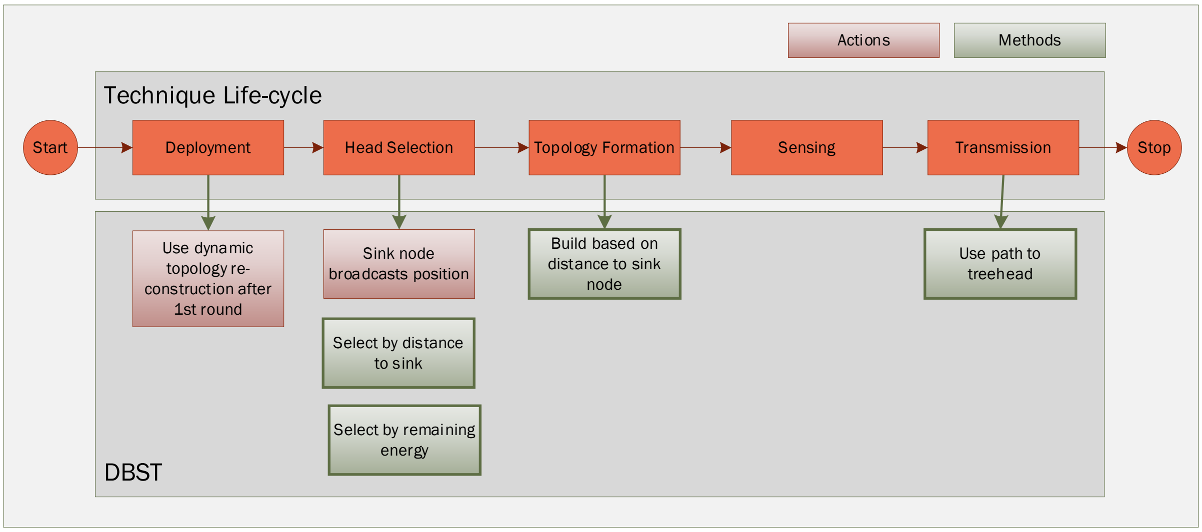

Figure 4-15 - Model design for DBST technique

Simulation and Results

The plots of Dbst is presented in figures 4.17. In comparison to other plots shown earlier, it presents a similar pattern of energy consumption behaviour based on the same WSN settings as used for the other techniques.

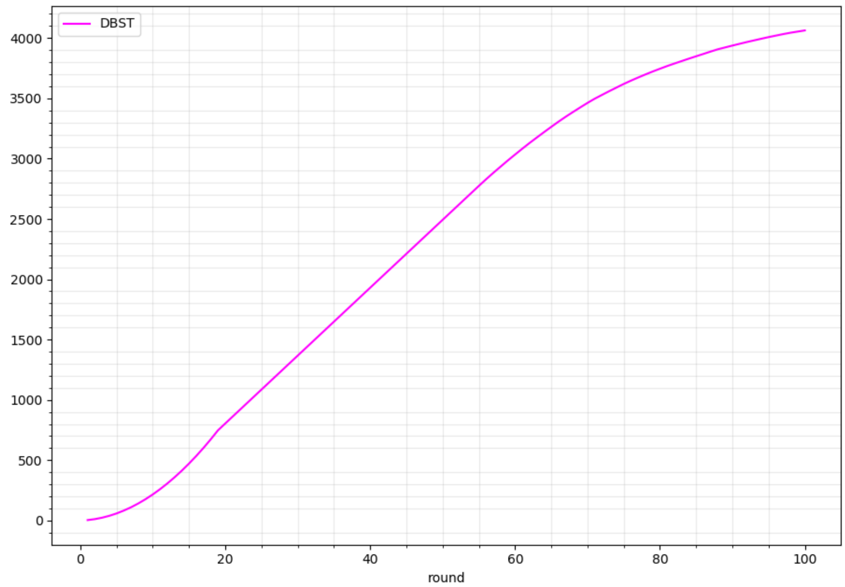

Figure 4-16 - Simulation plot of Dbst DAT on Energy Consumption vs Round for 100 nodes over 150 rounds

Directed Diffusion Technique

Although the Directed Diffusion technique is not considered in the final list of techniques used in this study, its performance is included in the following graphs simply as a reference plot compared to the other hierarchical techniques. It was noted that Directed Diffusion, being a Flat topology technique, is not included in this study.

4.8 Technique Behaviour Analysis

The discussions presented in sections 4.7.3, 4.7.4, 4.7.5, and 4.7.6 covered study of the behaviour of a set of techniques based on their energy consumption when run within a simulated WSN.. The following conclusions were drawn based on the results in the graphs presented:

1. While wireless sensor networks are in operation, various characteristics and attributes will hold different values, some of which are static, while others are dynamic, changing according to the progress of the event. Using the values of these attributes, it is possible to accurately evaluate the performance of the network, with respect to the application requirements and constraints.
2. DATs operate more effectively within a given range of values for some WSN attributes, while their performance can be changed based on the values held by such attributes.
3. Apart from the network (or WSN), the behaviour of data aggregation techniques is also impacted by other attribute settings, such as the scenario (which consists of the application running on the network).
4. Multiple techniques can be used interchangeably within the lifetime of a single WSN application, determined by the performance and driven by the objective and criteria of the running application. However, in order to ensure this possibility, a method is required to change the technique during the lifetime of the WSN application when the context is appropriate.
5. For a technique to be selected during a WSN's lifetime, it must be assured to be able to perform optimally under current conditions. This performance must be justified relative to other available techniques. In order to achieve this decision-making process, there needs to be a method to assess the technique's performance in near-real-time during the WSN operation, and to take decisions on the selection of a technique when considered optimal.
6. The system to detect the best technique needs to be context-aware in near-real-time and be capable of taking decisions given a set of values for specific attributes.

7. In order to improve its outcome, the sensing system must be able to learn from the outcome of its decisions and be capable of evaluating the best option to take given a history of prior outcomes and current system conditions.
8. An additional benefit and possibility would be to be able to incorporate more than one application within the WSN at a time. This would then require one or more techniques running at the same time when running applications require different techniques.

The following sections present more detail, which explain the relevance of these highlights. Each section focuses on one performance metric and provides more detail on how the differences in technique behaviour can be used to develop a decision-making mechanism.

4.8.1 Analysis of Energy Consumption Behaviour

This section discusses the behaviour of techniques based on the comparative network energy between rounds. A scenario is presented, which is modified to provide a new response from the techniques. The outcome, as shown in two different plots, is discussed in detail in a table where the specific values of energy consumption at each round is compared and the best value and technique selected.

The scenario used for this analysis involves a set of 100 nodes within an initial field size of 50 metres^2. The field size is then increased to 100 metres^2. A typical realistic scenario, which has some semblance to this use case involves a wildfire event. The event could spread into more areas, thereby increasing the required sensing field for the WSN. Some nodes could be turned off at the start of the event but turned on as the event spreads. The increase in field size could imply that nodes are having to transmit longer distances, thereby consuming more energy.

The two graphs shown in figures 4.18 and 4.19 illustrate the state of the network during the 50m^2 field size extending to the 100m^2 size. The rate of energy consumption as depicted in figure 4.19, is much higher than that depicted in figure 4.18. This is observable from figure 4.19 by the wider area under the graph after the energy levels off for each technique.

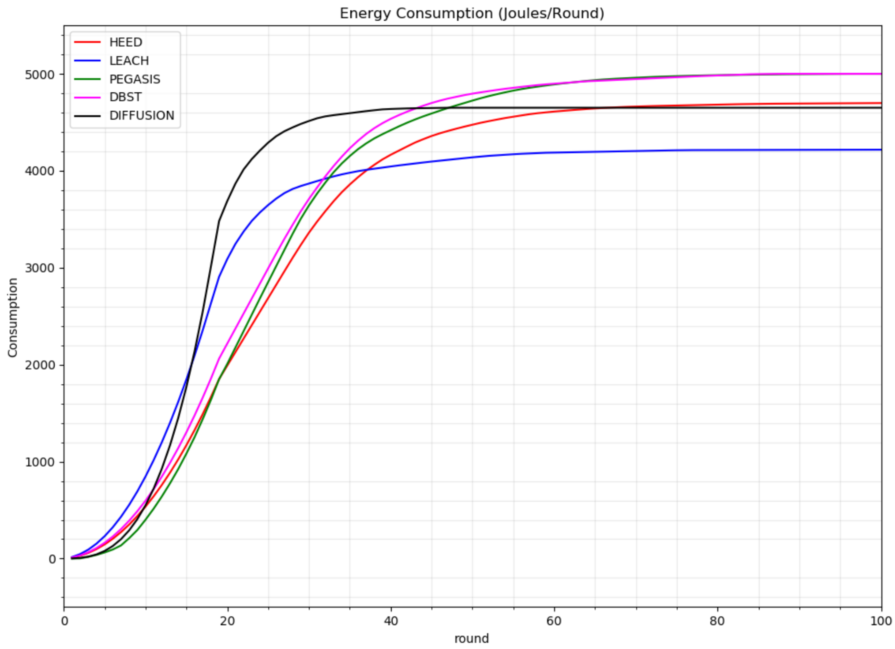

Figure 4-17 - Plot of energy consumption for selected techniques in a field size of 50m x 50m

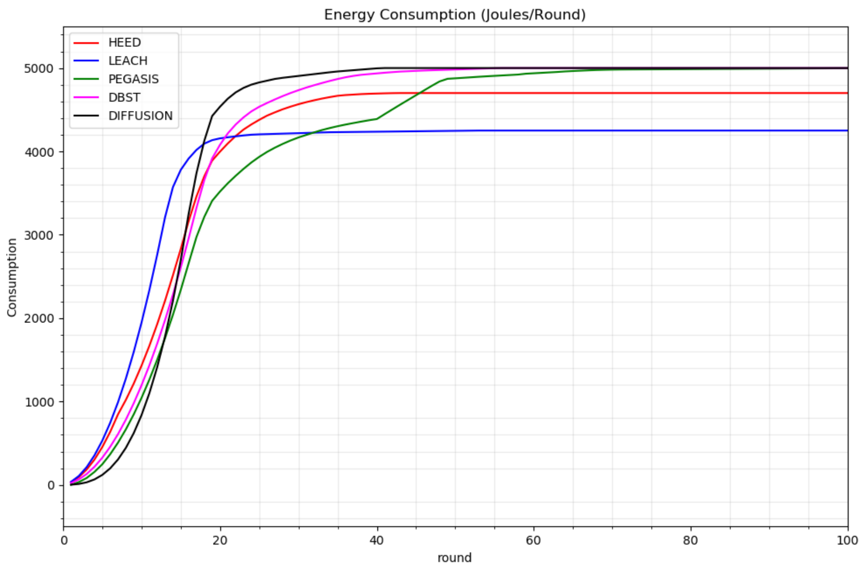

Figure 4-18 - Plot of energy consumption for selected techniques in a field size of 100m x 100m

The implication of the two graphs is that energy consumption changes due to certain environmental attributes. For instance, in figure 4.18, Directed Diffusion is observed to consume more energy in a field size of $50m^2$. In the simulation involving a field size of $100m^2$, the Leach technique consumes more energy than Directed Diffusion. Thus, the change in behaviour of the techniques based on the value of an attribute in the system supports the need to explore various approaches to optimise performance of the network using selected attributes.

According to on figure 4.18, it is observed that the Pegasis technique consumes the least energy between rounds 0 and about 20,. However, from round 21 up till about round 37, Heed consumes the least energy. Afterward, Leach seems to consume the least energy till about round 75, after which point its energy consumption flattens, perhaps due to the fact that the nodes can no longer transmit using their residual energy across longer distances caused by nodes that have turned off. After this stage, Heed takes on the best technique until about round 82, after which Dbst takes on the least energy consuming technique. Table 4.6 presents a selection analysis of the graph based on the best- and worst-case scenarios. The dash ("- ") indicates that the technique does not comply to the requirement of minimum energy consumption.

Table 4-6 - Selecting the best technique based on energy consumption by round taken from figure 4.18

No	Rounds	LEACH	HEED	PEGASIS	DBST	DIRECTED DIFFUSION
1.	2 - 20	-	-	Best	-	-
2.	21 - 37		Best	-	-	-
3.	38 - 75	Best		-	-	-
4.	76 - 100	-	Best	-	-	-

Table 4.7 presents the selection strategy for the best technique based on the energy consumption plot shown in figure 4.18. it indicates the selection of techniques based on the energy consumption in comparison to the remaining techniques. Table 4.7 provides more detail on the impact of selecting the best case against every other technique. The loss in percentage of selecting the best technique against each other technique is placed in a bracket as a negative number, after each energy consumption value in the form of XX (-YY%), where XX is the energy consumption of the technique at the specific round, and YY is the loss in percentage obtained by comparing the value of the least value technique to the technique with the XX value.

Table 4-7 - Analysis of the selection process of selecting the best technique based on energy consumption (in Joules) and loss encountered with the each of the remaining techniques (in bracket) with reference to figure 5.16.

Round	LEACH	HEED	PEGASIS	DBST	DIRECTED DIFFUSION
19	1616.35 (-42.911%)	1022.89 (-9.790%)	**922.75 (0%)**	1139.58 (-19.027%)	1449.13 (-36.324%)
21	2906.2 (-36.586%)	**1842.94 (0%)**	1849.7 (-0.365%)	2062.1 (-10.628%)	3481.54 (-47.065%)
38	**4009.35 (0%)**	4021.06 (-0.291%)	4282.07 (-6.369%)	4379.63 (-8.455%)	4615.04 (-13.124%)
76	No reading	**4543.67 (0%)**	4808.85 (-5.514%)	4845.14 (-6.222%)	4650 (-2.287%)
83	No reading	**4637.22 (0%)**	4927.11 (-5.884%)	4922.95 (-5.804%)	No reading

** the hyphen (-) indicates that the specific value for the technique cannot be used in the computation because the technique has stopped transmitting due to a loss of node energy across the network.*

In table 4.7, the selection strategy is shown with the values of energy consumption under each technique at each round. The gain obtained by selecting the better technique over the technique in a column is indicated in the bracket as a percentage. For example, at round 16 (falling between rounds 1 and 19), Pegasis would be a better choice to run the network on since it has the lowest level of energy consumption. However, at round 21 the best technique would be Heed. Back in round 16, the loss of selecting Pegasis over Leach would be 43% (shown in brackets). Thus, from table 4.7, it becomes obvious that it is beneficial to select the most optimal technique based on the energy consumption across the network. The next section provides a similar analysis based on the bandwidth consumption of the network.

4.8.2 Analysis of Bandwidth Behaviour

This section discusses the comparative behaviour of the techniques based on the bandwidth consumption per round. The same scenario used for the energy consumption analysis is used here. However, various WSN attributes were modified to obtain the responses, as shown in figures 4.20 and 4.21. As was the case for the energy consumption, the plots are discussed on a high-level in table 4.8, while a more detailed discussion is presented in table 4.9.

The field size used for the plot in figure 4.20 was $50m^2$, while that used for the plot in 4.21 was $100m^2$. This consists of a similar use case to the plots presented in figures 4.18 and 4.19, where field size is modified based on an event. The bandwidth can be impacted by channel loss leading to lost packets, and insufficient energy to transmit packets on the network.

The plot in figure 4.20 shows the bandwidth consumption for the selected techniques. Directed Diffusion consumes the maximum bandwidth due to its multi-node communication strategy. In terms of the preferred technique, the first technique that fits this requirement is Dbst, which remains the best technique, until about round 47. Afterwards, Heed is considered best as it consumes less bandwidth from that point on. Heed remains best until about round 80, when Pegasis becomes better than the rest. When this plot is compared to the plot in figure 4.21, a different selection process takes place. This time, Pegasis is considered the best technique at start. At about round 25, Heed becomes the best technique, and continues to remain so until round 80, after which Dbst takes on the best technique. Further discussions on this follow, based on data in tables 4.8 and 4.9, and using figure 4.20 as the reference plot.

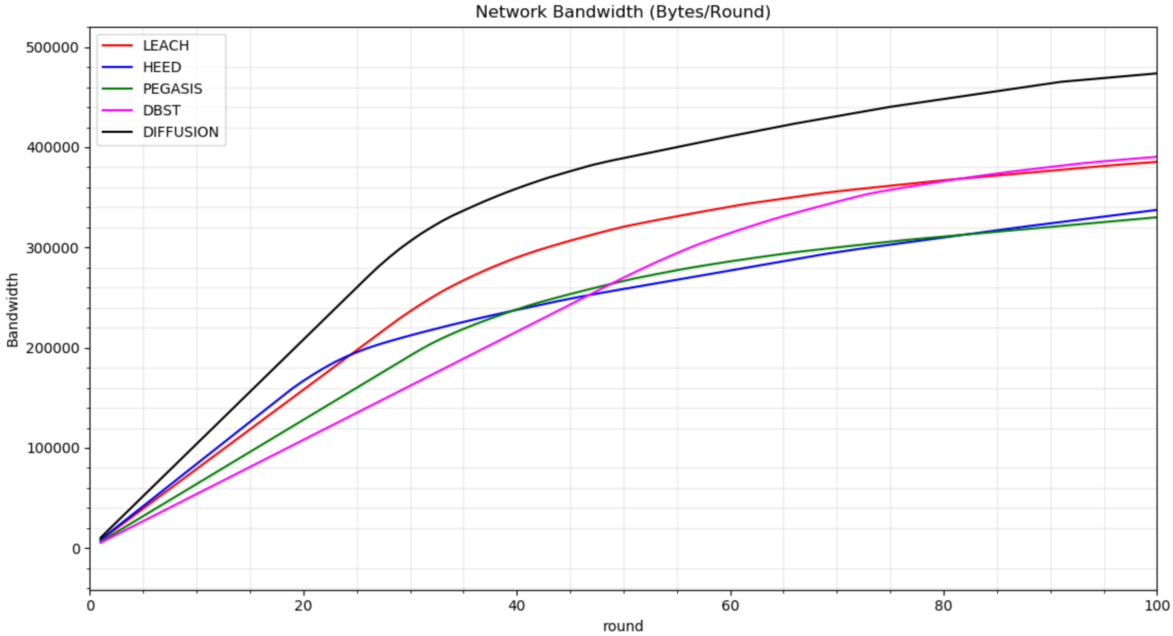

Figure 4-19 - Plot of Bandwidth consumption for selected techniques in a field size of 40m^2

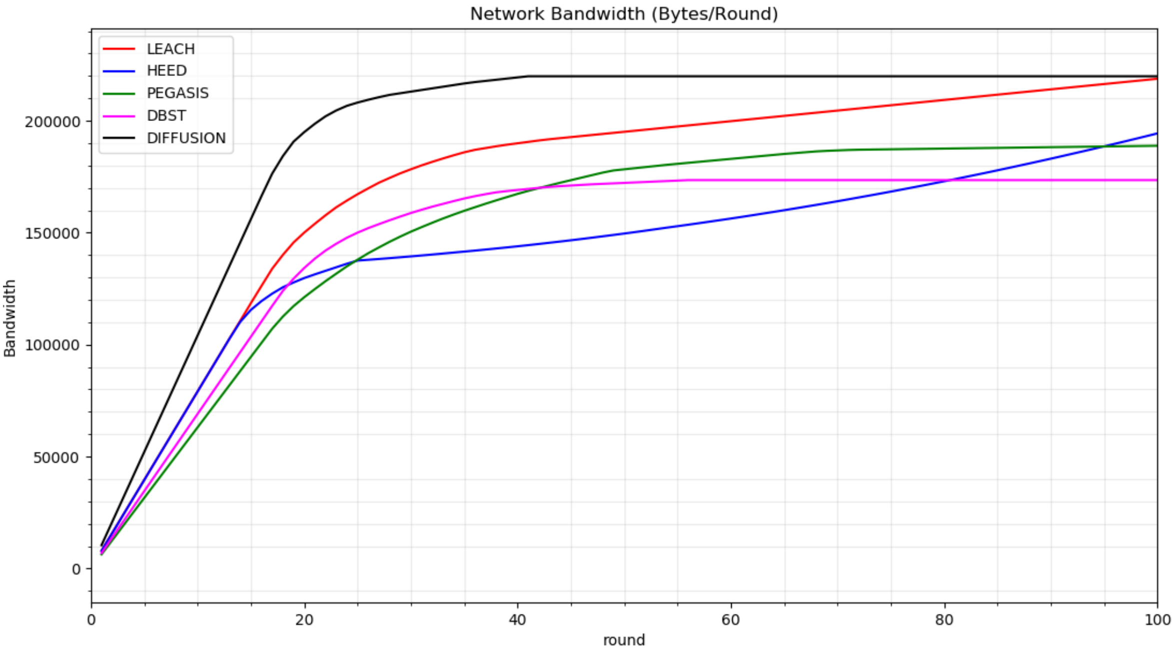

Figure 4-20 - Plot of Bandwidth consumption for selected techniques in a field size of 100m^2

From the two graphs, it can be observed that the behaviour of techniques based on their individual bandwidth consumption is different for different scenarios based on a change in characteristics, which in this case, involves the field size.

Table 4.8 presents the technique selection strategy based on the bandwidth consumption of selected techniques, as plotted in figure 4.20. The dash (-) indicates a technique that does not have interesting data considered enough for the purpose of this experiment.

Table 4-8 – Selecting the best technique based on bandwidth consumption by round taken from figure 5.4.20

No	Rounds	LEACH	HEED	PEGASIS	DBST	DIRECTED DIFFUSION
1.	2 - 48	-	-	-	Best	-
2.	48 - 82		Best	-	-	-
3.	82 - 100	-	-	Best	-	-

Table 4.9 illustrates the selection process for a technique based on the calculated bandwidth consumption plot shown in figure 4.20. The values of bandwidth consumption are indicated in the cell under the technique, while the loss involved in selecting other techniques are indicated in bracket as negative numbers under the appropriate technique.

Table 4.9 - Analysis of the selection process of selecting the best technique based on bandwidth consumption (in bytes) and the loss encountered with the each of the remaining techniques (in bracket) with reference to figure 4.20.

Round	Leach	HEED	PEGASIS	DBST	DIRECTED DIFFUSION
5	31600 (-12.658%)	30450 (-9.360%)	32000 (-13.750%)	**27600 (-0%)**	52000 (-46.923%)
48	309680 (-17.703%)	**254856 (-0%)**	256192 (-0.521%)	255800 (-0.369%)	379080 (-32.770%)
82	372801 (-16.473%)	318528 (-2.242%)	**311388 (-0%)**	375246 (-17.018%)	449696 (-30.756%)
98	384651 (-14.428%)	336168 (-2.087%)	**329152 (-0%)**	389610 (-15.518%)	472784 (-30.380%)

In table 4.9, the value of bandwidth consumption is used as a selection strategy. The comparison of each technique to the optimal technique at each time instant is also computed. This is indicated as the loss in brackets (-YY%) and represents the probable gain in bandwidth - though indicated as a negative due to its impact - by selecting alternative techniques other than the best technique.

4.8.3 Analysis of Latency Behaviour

This section covers the latency behaviour of the techniques during each round of the simulation. A similar scenario as used for energy consumption is applied here. The simulation is run first on a field size of 40m^2, and then on a field size of 100m^2. The plots generated from both simulations are shown in figures 4.22 and 4.23.

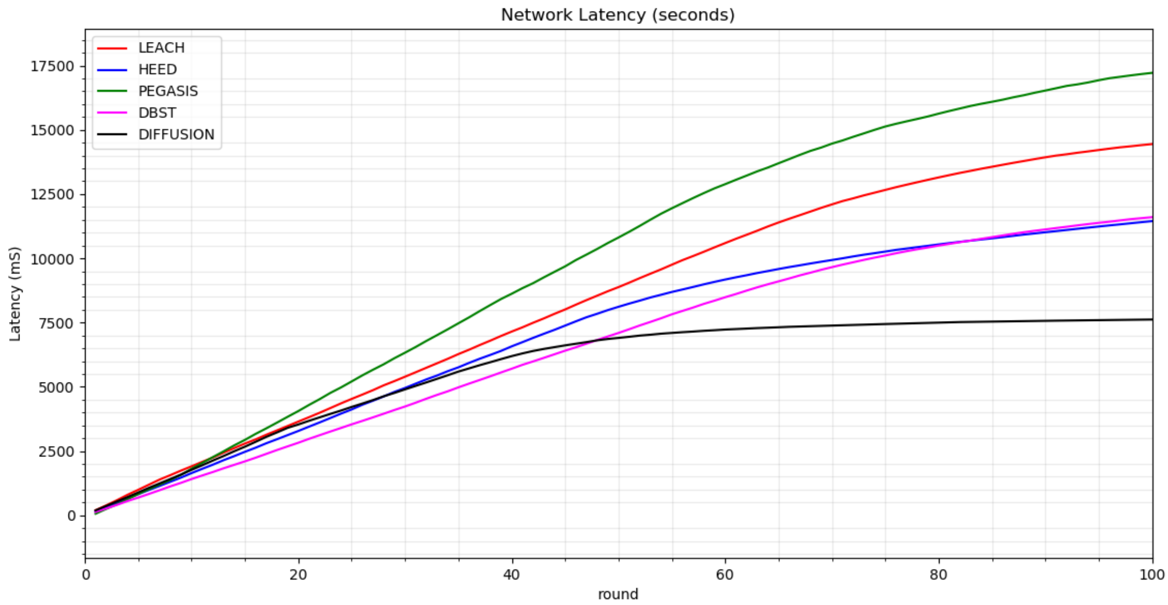

Figure 4-21 – Plot of latency for selected techniques in a field size of 40m x 40m

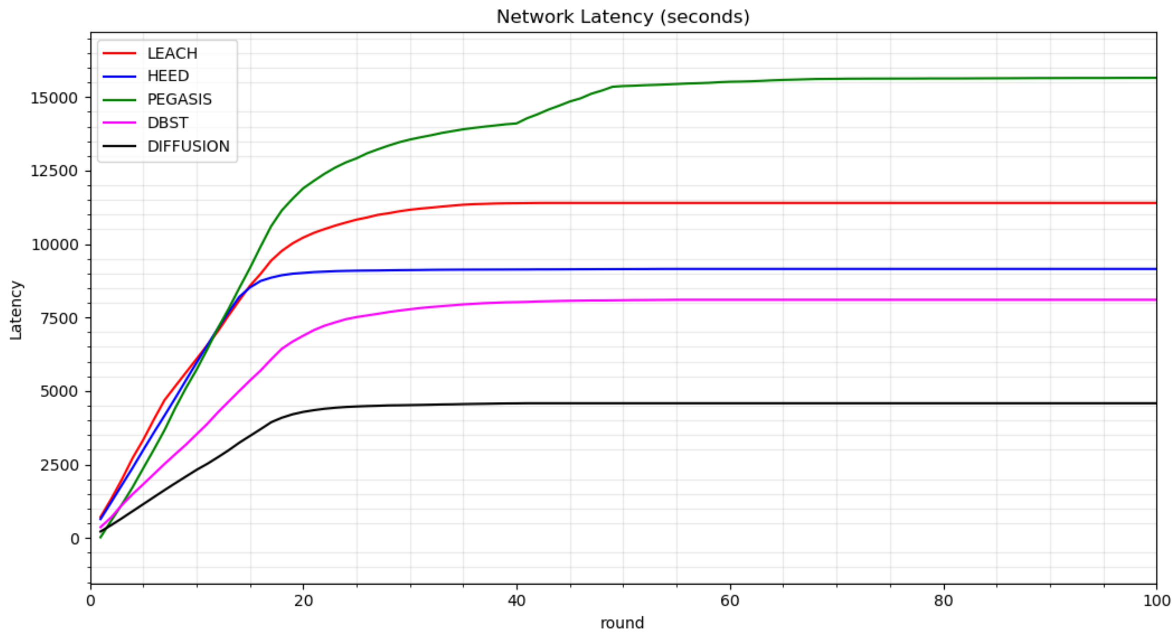

Figure 4-22 - Plot of latency for selected techniques in a field size of 100m x 100m

The differences between the two plots in figures 4.22 and 4.23 indicate that the latency experienced across the WSN with different techniques could be different based on different WSN settings. In figure 4.22, latency grows highest for Pegasis, while the lowest latency occurs for both Dbst and Directed Diffusion. While in figure 4.23, with a field size of $100m^2$, the Directed Diffusion technique has the highest latency.

Focusing on figure 4.22 the plot shows that Dbst started with the lowest latency of all the techniques, and this trend continues till about round 48, where Directed Diffusion takes over and continues till the end of the simulation. This behaviour and others are discussed further with the support of data in tables 4.9 and 4.10.

Table 4-9 - Selecting the best technique based on latency by round taken from figure 4.22

No	Rounds	LEACH	HEED	PEGASIS	DBST	DIRECTED DIFFUSION
1.	2 - 48	-	-	-	Best	-
2.	48 - 100	-	-	-	-	Best

Table 4.9 illustrates the selective preference of a technique based on its low latency performance in the network, while table 4.10 presents the same analysis as was done for energy and bandwidth consumption, this time, showing details for latency at specific rounds.

Table 4-10 - Analysis of the selection process of selecting the best technique based on latency (in msec) and the loss encountered with the each of the remaining techniques (in bracket) with reference to figure 5.22.

No	Round	LEACH	HEED	PEGASIS	DBST	DIRECTED DIFFUSION
1.	5	801 (-30.462%)	661 (-15.734%)	626 (-11.022%)	**557 (0%)**	889 (-37.345%)
2.	47	8547 (-21.856%)	7843 (-14.841%)	10365 (-35.562%)	**6679 (0%)**	6741 (-0.920%)
3.	90	14406 (-47.140%)	11449 (-33.488%)	17171 (-55.652%)	11564 (-34.149%)	**7615 (0%)**

Table 4.10 indicates that the best technique from the start was Dbst, considering latency comparisons from round 5. Thus, selecting Dbst at this stage would provide a gain of approximately 30% over Leach, for instance.

Based on the above analysis, it is shown that the best performing techniques can be selected from a pool of techniques given a performance objective.

4.9 Summary

This chapter has discussed various concepts which serve as a foundation for the further analysis, experiments and development that continue in the following chapters. The chapter discussed attributes and variables according to the definitions required for this study. It also introduced the concept of WSN dimensions, which can be used to distinguish WSN techniques as well as evaluating their performance. The main WSN components consisting of the Technique, Network and Scenario, were introduced, while their relationships with the WSN dimensions were also discussed, including how a relationship workflow could be used as a guide in applying their interrelationships towards analysing a WSN application. Beyond these, the chapter also discussed WSN models for the selected techniques. The chapter also covered a behavioural analysis of the techniques using complex scenarios and presented the fact that based on the specific objective of a WSN application, there is most often a technique that performs better than the rest. These concepts shall be used as a foundation for further discussions in this thesis.

The next chapter discusses further formal analysis of the various concepts introduced so far in the study, including further discussions about the simulation environment.

5 WSN Data Aggregation Recommender Machine Learning Model Analysis

5.1 Overview

This chapter covers the formal analysis of the concepts introduced in past chapters. In discussing these, various aspects of the system are presented while their relationships are discussed using mathematical models.

The chapter first presents discussion on relevant concepts presented in earlier chapters, such as *Attributes*, *Techniques*, *Network* and *Scenario*. The interaction between these components is revisited to emphasise their relevance to discussions in this chapter. A huge section of the chapter focuses on the formal definition of various components using mathematical equations to highlight their relationships. This definition step provides input into a more holistic analysis of the intelligent algorithm, which applies the components to determine various factors which are considered important for its decision-making process. The intelligent algorithm involves the data processing stage of the system, where the optimal technique is selected, whose outcome is pre-processed ahead of being used to train the machine learning model. The chapter then discusses the architecture of the intelligent model, and this is used to illustrate the workflow of the system, serving as a reference point for further analysis and development of the model. The "*reference*" architecture is also used to highlight the components that implement various equations based on the formal analysis performed earlier.

Following this, further design of the intelligent model is discussed. This involves the review of various attributes that have been presented in past chapters including their corresponding data types. On determining the shape of the input for the machine learning model, the architectural details of the model are discussed and presented based on similar machine learning experiments for WSNs in the literature. The implementation of the architecture is discussed, and its initial hyper-parameters are provided.

5.2 Overview of Relationship Between WSN Components and the Intelligent Model

Figure 5.1 provides an overview of the relationship between the WSN components and the intelligent model and provides a high-level flow between the components. The main areas are delineated by the coloured green and red boxes. The green box encompasses the components that have been introduced earlier and discussed in detail in chapter 4 and illustrates how the components are combined to form the feature definition and subsequently, the feature extraction process. The green box contains the primary WSN components, which include the *Technique, Network and Scenario*. Each component has its own set of attributes, which are also classified based on various variables. The green box helps to define the expected shape of the data that will be provided as input to the red box. The red box encompasses elements that are discussed, designed, and implemented in this chapter, and showcases the high-level representation of the intelligent algorithm and machine learning model.

The green box essentially represents the *feature extraction process*, which involves the determination of the form of data that are necessary for the algorithm. The red box, consisting of the rule-based algorithm and intelligent model, applies rules to determine data input for the intelligent model. The input of the algorithm consists of data about application requirements (application objective(s), criteria, and weighting), a set of techniques, a network component, a scenario component, and their associated attributes. The resulting data serves as training data for the intelligent model. This is discussed in more detail in subsequent sections. The intelligent model is trained based on this data. The recommendations of the intelligent model (ML model) are stored within a knowledge base and are used as new training for the ML model.

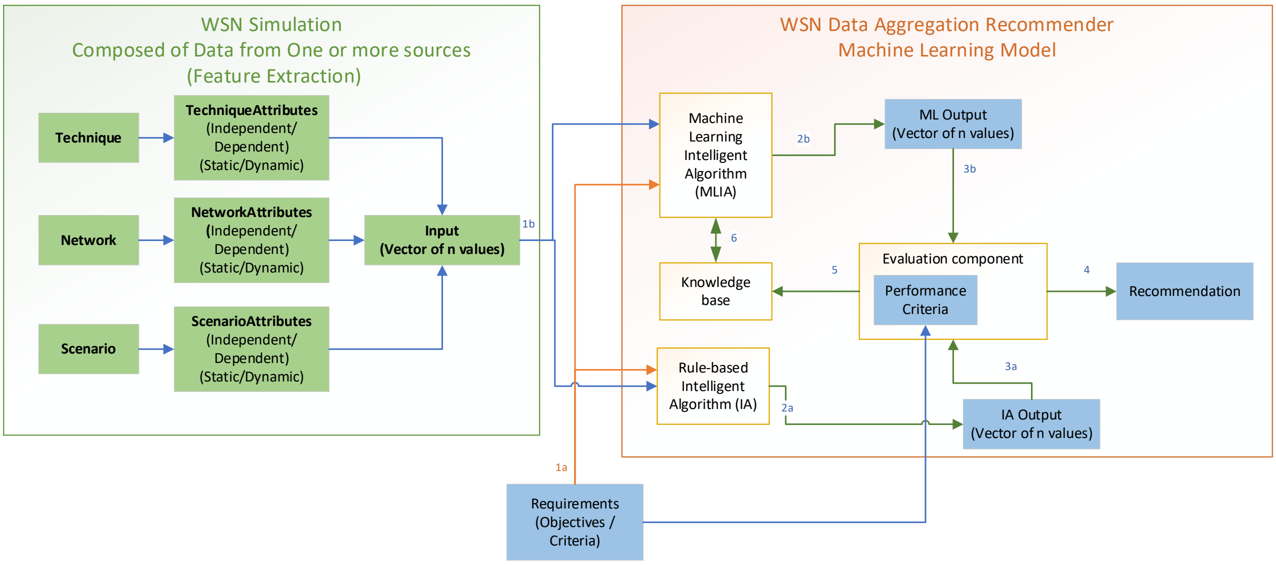

Figure 5-1 - High level overview of the relationship between the concepts of WSN components and the intelligent machine learning model. The numbers indicate the flow of the process. It starts from the output coming out of the green box. The lines differentiate the environment of the process flow, i.e. Feature Extraction, Requirements Input, Rules-based Algorithm, and the Intelligent Machine Learning Model.

The workflow presented in figure 5.1 is discussed below:

1a and 1b: Data consisting of techniques, scenarios, and networks are compiled and combined with application requirements, which consist of the objectives, criteria and priority. These are provided as input into the rule-based algorithm. The outcome of the algorithm consists of the best technique based on the data submitted.

2a: The input data used to perform the decision consists of the application requirements, with various environment settings (represented by the network and scenario components) and serve as input into the intelligent machine learning model (ML model). The output consists of the recommended technique based on the input data.

2b: The machine learning model provides its recommendation in the form of a vector of binary values, the number of which is defined by the set of techniques under consideration.

3a: The rule-based algorithm is used as an evaluation tool to validate the output of the ML model, enabling the ML model's recommendation to be compared with the ground truth.

3b – The ML model's recommendations are channelled into a component that enables some form of evaluation, and which continuously measures the accuracy of the prediction. This can be used to obtain a measure of accuracy of the model over time.

4 – The ML model's prediction is provided as a recommendation based on the submitted query.

5.3 Description of the Reference Architecture

This section discusses a process that covers the steps used throughout the study, which includes the initial investigation, up to the development of the intelligent model. The stages, shown in figure 5.2, are identified as (1) research and design, (2) identification of system components (such as technique, network and scenario), (3) identification of features, (4) definition of data collection requirements, (5) design of data pre-processing steps, (6) rule-based algorithm design and development, (7) definition of input and output vectors for intelligent model, (8) data generation for model training, (9) data processing through the rule-based algorithm, (10) intelligent model training, (11) intelligent model evaluation. These steps are illustrated in figure 5.2.

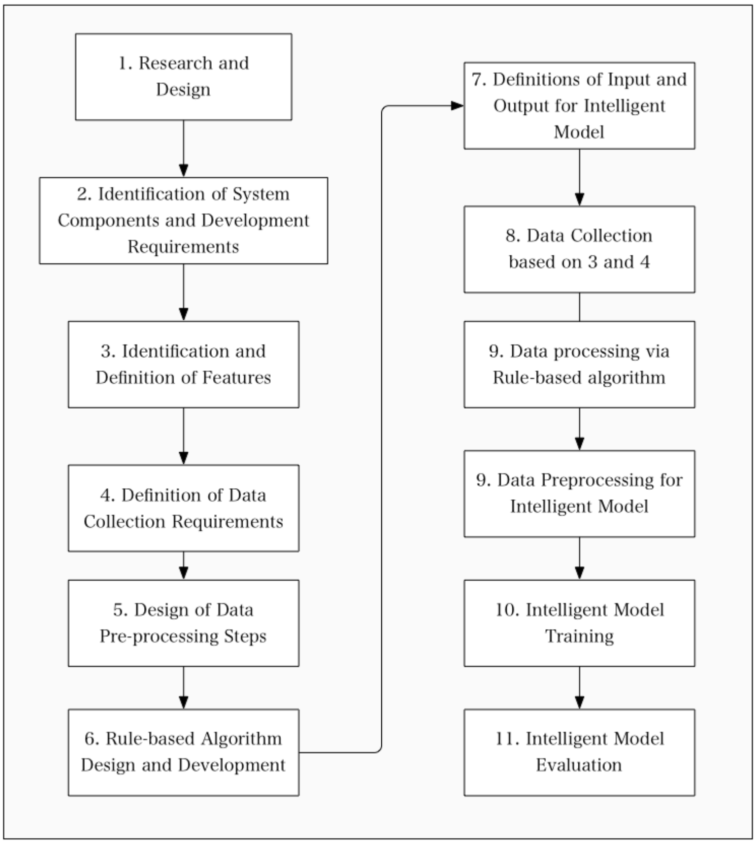

Figure 5-2 - Process flow for study starting at initial investigations, up to the intelligent model development.

The following sections cover the formal definition of the WSN components involved in the study.

5.4 Formal Discussion of WSN Components

This section provides a formal analysis of the system consisting of the intelligent algorithm and the machine learning model. This analysis serves as input into the development of the intelligent algorithm and the machine learning model. The content in this section consists of the discussion of the symbols used in the various mathematical equations, which are later used to describe the relationships between various components of the WSN model. Such relationships are later used to define the computation of the various performance metrics. These discussions are later used to define other equations, which illustrate the workings of the intelligent model.

The discussion commences with the WSN model, and progresses into attributes and the WSN components, including their sub-component attributes. Afterwards, it proceeds to discuss the WSN as a system consisting of one Scenario, one Network and multiple in-active Technique components. Finally, the relationships are used in defining the optimization goals of the intelligent model based on various equations. Table 5.1 provides a reference for the collection of symbols used within the equations.

Table 5-1 - Symbols used in the formal analysis discussions

Symbol	Definition	Description
W	Wireless sensor network	This represents the wireless sensor network model in consideration.
A	WSN Attribute	This represents an attribute as discussed in section 3.3.2
T	WSN Technique	This represents a technique entity, as discussed in section 4.4 as one of the primary components of a WSN model.
S	WSN Scenario	This represents a scenario entity as discussed in section 4.4 as one of the primary components of a WSN model. The scenario encapsulates the specific WSN application context outside the network.
N, K	WSN Node, Network	N represents a network entity as discussed in section 4.4 as one of the primary components of a WSN model. However, in certain equations and discussions, N could be used to represent a node as well. When this is the case, K is used to represent the network of nodes.
T_D	Technique domain	This represents the collection of techniques that are available during a WSN simulation.
S_D	Scenario domain	This represents the optional scenario contexts in which the WSN model can be placed during the simulation.
N_D	Network domain	This consists of the domain of network entities and represents optional network settings that can be applied to the WSN model during a simulation.
G, V, E	Graph, Vertex and Edge.	Directed graph, vertices, and edges of the graph. These are used to provide a theoretical description of the structure of the WSN and are only used for this purpose and not applicable in further discussions beyond that point. Please refer to Appendix C for a more detailed introduction to Graph Theory
R	Rounds	This represents the total number of rounds performed in a WSN simulation, comprising of several rounds each represented by r.

BS	Base station	This identifies the final target for communication and could be physically represented by a gateway or sensor node. It is usually located in proximity to the network, either within the network or external to the network. It could also be responsible for the final stages of data aggregation.
L	Size of one edge of a virtual square field region	This represents the length of a field region. A square area is used as the default field size where nodes are deployed. Thus, an *L* value of *50* implies a *50m x 50m* wide field area.
I/D	Independent / Dependent attribute	This represents the types of attributes defined as dependent and independent attributes.
A_v	Attribute value	This represents the value held by an attribute.
T_v	Type of an attribute value	This defines the type of the value held by an attribute, indicating either real, ordinal or binary value type.
W_v	Attribute weight	This holds the weight assigned to an attribute and is applicable in the specification of WSN application objectives and criteria.
Th_v	Attribute value threshold	This represents the threshold assigned to an attribute (especially of a metric), which determines its cut-off point based on whether it is being minimized or maximised.
M_{min}, M_{max}	Maximum and minimum threshold values	These represent the maximum and minimum value that can be held by certain attribute value types such as those holding real numbers. This is also applicable in the specification of WSN application objectives and criteria.
r	Round	This represents the round in a WSN simulation.
O, C, P	Application Objective, Criteria, Priority	These represent the set of application objective(s), criteria and set of priorities, which are used to determine the appropriate technique that fits the context and requirements.
E	Energy consumed	This represents energy consumption. This could be by a technique (*T*), a node (*N*), or the network (*K*).
dE	Change in energy consumption	This represents the change in energy consumption across two rounds. This can be measured on a node by node basis, or across the network.
B	Bandwidth consumed	This represents bandwidth consumption. This could be by a technique (*T*), a node (*N*), or the network (*K*).

dB	Change in bandwidth consumption	This represents the change in bandwidth consumption across two rounds. This can also be measured per node or across the network.
n, m, p	Instance counter	These are used to specify count of instances of an object in a collection, such as the number of nodes.
i, j, k	Index counter	These are used to perform counting in an enumeration, such as indicating unique instances of an object class.

The symbols presented in table 5.1 are used in the equations discussed in sections that follow, and thus, can be used as a reference point in understanding the definition of the symbols.

5.4.1 WSN Model

The WSN model used in this research is formally defined as consisting of a network topology represented as a directed graph $G = (V, E)$, where V represents the vertices or set of nodes, and E represents the edges or communication links between the nodes. The *Graph Theory* on which this discussion is based is discussed in more detail in *Appendix C*.

The physical topology consists of n number of sensor nodes, with each node having computing resources, memory and networking. The nodes are randomly distributed in a square region defined as LxL, where L is in metres, and a base station (*BS*), such as a gateway, located outside the perimeter of the network.

The instance of a WSN model is represented as W_i, and contains instances of the three primary components, that is *Scenario* (S_i), *Network* (N_i), and a collection of *Techniques* ($T_1, \ldots T_n$). The WSN can be considered to compose of a collection of attributes ($A_1, \ldots, A_n$), all defined under one of these components. This enables a WSN to can be represented as either of the two equations:

$$W_i = \{ A_1 \ldots . A_n \} \quad \textit{Equation 1}$$

$$W_i = \{S_i, N_i, \{T_1, \ldots, T_m\}\} \quad \textit{Equation 2}$$

where i is an index identifier for the WSN, W_i is the WSN instance, $A_1 .. A_n$ represent the collection of attributes, with n being the total number of attributes. In equation 2, S_i, N_i, and T_i all represent instances of the scenario, network, and techniques respectively, while the value of i in each instance does not indicate any relationship. Also, $T_1 .. T_m$ represent the set of available techniques in a WSN simulation, where m represents the total number of techniques. Both equation 1 and equation 2 represent two alternative approaches to stating the relationship between the WSN and it's set of attributes. The discussions presented in this chapter (and this study) use the form presented in equation 2. The next section discusses the *Attribute* of the WSN.

5.4.2 Attribute Component

An *Attribute* defines the smallest component of the WSN. All other components are built based on a set of attributes. An attribute consists of various components, which can hold various values throughout the lifespan of the WSN application. The WSN attribute consist of a set of parameters as indicated in equation 3:

$$A_i = \{ v, T_v, W_v, Th_v, MIN_v, MAX_v, D, S\}$$ *Equation 3*

where v represents the value, v_t refers to the type, v_w refers to the weight, and v_{th} refers to the threshold. The parameters d and s represent the attribute's dependence and static/dynamic states respectively. M_{min} and M_{max}, both representing minimum and maximum values for the attribute, are relevant when the attribute is used as an application requirement. Thus, when the attribute v_t indicates a real number, M_{min}, and M_{max}, will hold the extreme values for the value in v_v, thus providing a criterion for the attribute's value. Similarly, v_{th} and v_w will both hold values that define how much the impact of including the attribute in computation affects the outcome. Using the first letters of the states, v_t can hold one in the set of states for the lifespan of a WSN application, as indicated in equation 4.

$$v_t = \{ O \mid R \mid B \}$$ *Equation 4*

where O represents *Ordinal*, R represents *Real*, and B represents *Binary*. Also, in another dimension, an attribute can either be dynamic or static, and either dependent of independent. These two classifications are represented in equations 5a and 5b. The concepts of *Dependent* and *Independent* in equation 5a relates directly to discussions on the same concepts held in section 4.4.2 and discussed earlier in this chapter. The concepts of *Static* and *Dynamic* as presented in equation 5b also determine whether v_v of the attribute changes or remains static throughout the lifespan of the WSN application (or applications). These were also discussed in section 4.4.2 and reviewed earlier in this chapter.

$$v_d = \{ I \mid Da \}$$ *Equation 5a*

$v_s = \{ S \mid Db \}$ *Equation 5b*

where *d* represents the attribute's dependent state with respect to other attributes, *s* represents the attributes static/dynamic state. In equation 5a, *I* represents *Independent*, *Da* represents *Dependent*, while in equation 5b, *S* indicates Static, and *Db* represents *Dynamic*. Thus, an attribute must have both v_d and v_s defined based on the options indicated in equations 5a and 5b. The next section discusses the three primary components of the WSN model, i.e. the Technique, Network, and the Scenario. These components are defined by a specific set of attributes that define their characteristics.

5.4.3 Technique Component

The technique referred to in this section corresponds to the technique definition presented in section 4.4 and indicates the same technique discussed earlier in this chapter. The technique represents a single WSN data aggregation technique T_i, one instance in the set of techniques, which are available for selection during a WSN application. The corresponding equations are defined in relation 1 and equation 6 below.

$T_i \in T_D$ *Relation 1*

$T_i = \{ A_{i1}, A_{i2}, \dots, A_{in} \}$ *Equation 6*

where *i* is an index used to identify a technique instance T_i, and T_D represents the techniques domain.

Equation 6 defines a technique composing a set of attributes $\{A_{i1}, \dots, A_{ni}\}$, where A_{i1} refers to the first attribute of T_i, and *n* indicates the total number of attributes of the technique. The value held by these attributes determine the behaviour of the technique while operating within a given WSN application. The process involved is discussed in more detail in further sections. The next section discusses the *Network* component.

5.4.4 Network Component

A *Network* instance, K_i, represents one of the primary components of the WSN, and is an element in the domain of networks K_D. The network concept was introduced in section 4.4 and earlier in the chapter. The possibility of changing network attributes is determined ahead of a WSN application event and should determine if there are more than one network instance. The network domain and attribute relationship are both defined in relation 2 and equation 7 below.

$$K_j \in K_D \qquad \textit{Relation 2}$$

$$K_j = \{ A_{j1}, A_{j2}, \dots, A_{jm} \} \qquad \textit{Equation 7}$$

where j is used as an index to represent an instance of a network K, and K_D is the domain of Network instances. Equation 7 defines the relationship between the network K_i and the set of attributes, where A_{j1} represents the first attribute of the set attributes for network K_i and m indicates the total number of attributes. As discussed earlier, the combined values of these attributes define the state of the network instance K_j.

5.4.5 Scenario Component

The *Scenario* represents the third main component in the WSN model, an element in the scenario domain S_D, where S_D represents the domain of scenarios. This term relates to the same concept as introduced in section 4.4 and discussed earlier in this chapter. This component encapsulates the details of the environment of the WSN to enable manipulation during the WSN event. This contributes to the specification and comparison of application use cases. Its domain relationship and attribute relationships are shown in relation 3 and equation 8.

$$S_k \in S_D \qquad \textit{Relation 3}$$

$$S_k = \{ A_{k1}, A_{k2}, \dots, A_{kp} \} \qquad \textit{Equation 8}$$

where k is used as an index to identify the scenario instance S_k, S_D represents the domain of scenarios, A_{k1} represents the first attribute in the set of attributes for the scenario instance, and p represents the total number of attributes.

Equations 6, 7, and 8 are needed to represent the main components within a WSN application. They are used together to define the running network and to be able to determine the impact of the network on the selection of the best technique for optimum performance. However, the relationship between the attributes across the various components needs to be discussed further and this is done in the next section.

5.4.6 Similarity in WSN Entity Attributes

According to equations 6, 7 and 8, the three WSN components, i.e., *Technique*, *Network* and *Scenario,* have sets of attributes that are part of a larger collection of attributes. The attributes chosen by each component do not have any relationships across component boundaries. Event though, a minimal number of cases exist where components have attributes with similar names, their function is isolated under the component. The only occasion where a relationship exists is with techniques, which all share the same set of attributes. The values of the attributes contribute to evaluating the performance of the technique. The attribute relationship equations for the technique and the network are re-represented below.

$$T_i = \{ A_{i1}, A_{i2}, \dots, A_{in} \} \qquad \textit{Equation 6}$$

$$K_j = \{ A_{j1}, A_{j2}, \dots, A_{jm} \} \qquad \textit{Equation 7}$$

where A_{i1} in equation 6 represents the first of the attributes for technique T_i, and A_{j1} in equation 7 represents attributes for network K_j. Based on the discussion on similar attributes across components, if $A_{i1} = Aj_1$, there is not meaning attached to this equation when it occurs during the lifespan of a WSN application. The attributes are defined under the entity that owns them and are not transferable or replaceable as the corresponding attribute under another component. Likewise, the number of attributes for both components, i.e., *n* and *m*, could hold the same value but do not, in such cases, imply any meaning.

5.5 Discussion on WSN Metrics

5.5.1 Definition of Energy Consumption

The WSN energy consumption model's corresponding equations are presented in this section. The derivation of the essential component-to-energy equations are discussed, while these are used to define the optimal value equations for the energy consumption metric.

As a reminder for discussions presented here, the WSN can be assumed to consist of a network *K* consisting of *n* nodes, each labelled as N_i. where *i* represents the identifier for the node *i*. To calculate the cumulative energy consumption of the nodes within a network during a simulation, otherwise referred to as the network energy consumption, the following equation can be used:

$$K_E = \sum_{i=1}^{n} N_{iE} \qquad \textit{Equation 9}$$

where E represents energy, K_E represents the network energy consumption, i is used as an identifier for each node instance, and n represents the counter of the number of nodes involved in the network. Thus, N_{iE} represents the energy consumed by a node instance N_i. This equation applies to a round and measures the total energy consumed by all nodes at the end of the round. However, the energy consumption in round r can be clearly defined as K_{Er}, where r is used as an identifier for the round. Based on this and the discussions presented so far, K_{Er} will be different for different values of r, and different based on the technique applied in the simulation. The energy consumption for a technique T_i in round r can be represented as:

$$T_i = K_{iEr} \qquad \textit{Equation 10}$$

where the i is used as an index identifier for the technique instance, E is energy, r is round and K_{iEr} represents the sum of energy consumption for the technique T_i in round r. The energy represented by K_{iEr} needs to be minimized. Taking into consideration all possible rounds R, the total energy consumption in a simulation by a technique can be calculated as a sum of the consumption in each round r, and be defined as follows:

$$T_{iE_R} = \sum_{r=0}^{R} (K_{iEr} - K_{iE(r-1)}) \qquad \textit{Equation 11}$$

where T_{iE_R} represents the energy consumption for technique instance across R rounds, K_{iEr} represents the energy consumption in round r for technique with identifier i, and $K_{iE(r-1)}$ representing the previous round. Based on prior arguments the best technique proposed by the system for each round r, i.e. T_{Br}, will change as the context of the WSN changes throughout the lifespan of the application. The best technique after round r would be determined by comparison based on the application of equation 10 to all available techniques after round r, in order to obtain T_{Br}. If the change in energy consumption is represented as dK_{Er}, then equation 11 can be re-written as follows:

$$T_{iE_R} = \sum_{r=0}^{R} dK_{iEr} \qquad \textit{Equation 12}$$

which then represents the summation of energy consumption for a technique T_i in each round r of the simulation, which runs a full length of R rounds. Typical values obtained for this equation were presented earlier and plotted for five techniques back in chapter 4 in figure 4.18.

Assuming that four techniques are available for selection within a simulation of R rounds, each technique would have its own energy consumption for each round based on equation 12. The data generated per technique for similar runs across the same network settings (including rounds) will be collated pre-processed and passed as input into the intelligent model. Figure 5.3 provides an illustration of the process used by the system to determine the best technique during a WSN application event.

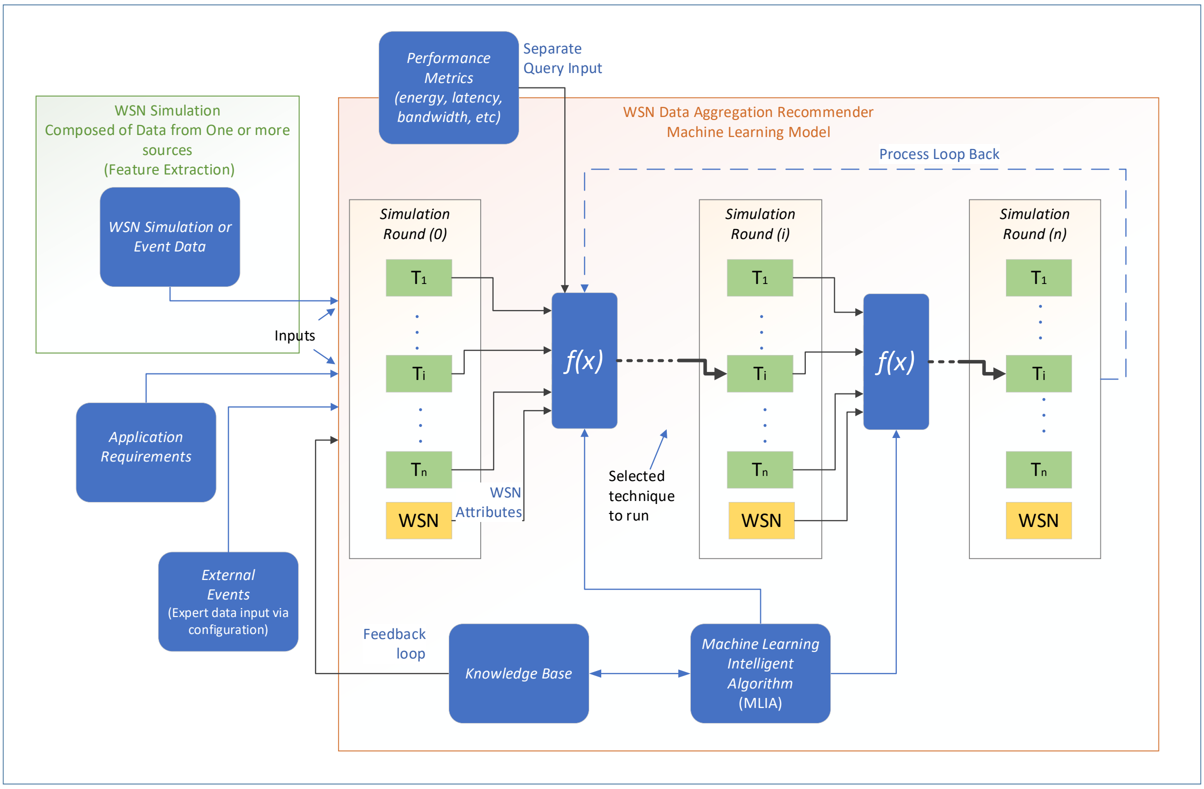

Figure 5-3 - High-level illustration of the entire system integrated with the intelligent model for providing recommendations. It shows the inputs, computation, output, knowledgebase, and feedback loop.

In figure 5.3, the box with symbol *f(x)* represents the machine learning intelligent algorithm (MLIA), which performs the selection and learning process. The green boxes indicate the set of techniques in the simulation, while the black lines connecting the green boxes to the *f(x)* boxes contain the attribute values being passed to the intelligent model. The intelligent model combines the application requirements, consisting of the objectives and criteria, with these to compute the best technique suited to the current state of the simulation. A feedback loop exists to provide an avenue to feedback the recent changes to the intelligent model. The integrated knowledge base can be used to determine the results of past recommendations before querying the ML model. Once the ML model is sufficiently trained, it is used to recommend techniques based on provided data.

Continuing further with the formal analysis, based on equation 12, the function *f(x)* needs to process a set of performance metric quantities, in this case, energy consumption, as defined by equation 13.

$$f(x) = \{ T_i dK_{Er}, T_{i+1} dK_{Er}, T_{i+2} dK_{Er}, T_{i+3} dK_{Er}, T_{i+4} dK_{Er} \} \qquad \textit{Equation 13}$$

The machine learning intelligent algorithm (MLIA or intelligent model from here on) takes into consideration much more metrics than just energy consumption, including the requirements, such as objectives, criteria and priorities. Other metrics are discussed in the following sections. The application requirements input consist of the following:

1. Objective(s) (e.g., minimize energy consumption)
2. Criteria (constraints such as minimum/maximum values for metrics, maximum sensor node energy, etc.)
3. Priority (ordering of metrics at decision point, defined as using the weight value assigned to metric attributes)

Assuming that these requirements are symbolised as a set of Objective(s) – $\{O_i, ..., O_n\}$, set of criteria - $\{C_1, ..., C_n)$, and a list of priorities – $\{P_1, .., P_n\}$, then equation 13 an be exploded as defined in equation 14.

$$f(x) = \{\{O_1, .., O_n\}, \{C_1, .., C_n\}, \{P_1, .., P_n\}, T_{Er}, K, S\} \qquad \textit{Equation 14}$$

where, $T_{Er} = \{ T_i dK_{Eir}, T_{i+1} dK_{E(i+1)r}, T_{i+2} dK_{E(i+2)r}, T_{i+3} dK_{E(i+3)r}, T_{i+4} dK_{E(i+4)r} \}$, *K* represents the network, *S* represents the scenario, and the others (application requirements) are as discussed above.

Computation of Minimal Values

The goal of the function defined in equation 13 involves obtaining optimal values for the specified objectives. In order to reduce the complexity of the calculations, a single technique, T_i, is used in this discussion. According to equation 6, T_i has a set of attributes $\{A_1, ..., A_n\}$, each having a value and type. If an attribute of energy consumption is selected. This attribute could be selected to also serve as a metric to evaluate the technique. In the application requirements, a criteria definition includes the acceptable range of values defined by extreme values M_{min} and M_{max}. With this, the function in equation 15 would use the following definition to compute the compliance of a technique based on the requirements:

$$f(t) = \{ T_i dK_{Er} < X_{max} \mid T_i dK_{Er} > X_{min} \} \qquad \textit{Equation 15}$$

where T_i represents the technique, dK_{Er} represents the energy consumption after round r, X_{max} and X_{min} both represent the minimum and maximum extremes for the attribute.

In the normal case, the definition of equation 14 applies to all real number attributes, and for all participating techniques, throughout the lifespan of the WSN application.

5.5.2 Definition of Bandwidth Consumption

The evaluation of bandwidth consumption is performed in a similar sense to that done for energy consumption. Bandwidth is a typical technique attribute, which also qualifies as a metric to evaluate the technique's performance.

Assuming a WSN which consists of a network K of n nodes. The bandwidth consumed by the network consists of the number of bytes transmitted during communication, from the sensor node, across other nodes, towards the sink node. Other variables that are not included in this discussion, but which affect the results in the simulation, include the channel's data rate – the rate at which bytes are transmitted, and path loss – the loss function of the channel, which determines how many packets are lost in transit. The bandwidth consumption for each node can be defined as in equation 16:

$$K_B = \sum_{i=0}^{n} N_{iB} \qquad \textit{Equation 16}$$

where K_B represents the total bandwidth consumption across the network in a given time, i represents an index identifier for the nodes on the network, n is the number of nodes, and N_{iB} represents the bandwidth consumed by node instance N_i. This equation can be used to represent bandwidth consumption in a round using K_{Br}. A new value is generated for K_{Br} at the end of every round, with the value increasing in subsequent rounds. The bandwidth consumption for a technique in round r can be defined as:

$$T_{iBr} = K_{Bri} \qquad \textit{Equation 17}$$

where the T_i represents the technique instance, T_{iBr} represents the techniques bandwidth consumption in round r, and K_{Bri} represents energy consumption across the network for round *r*. Based on this relationship, the cumulative bandwidth consumption of a technique across a WSN simulation, over a set of rounds *R*, can be defined as follows:

$$T_{iBR} = \sum_{r=0}^{R} (K_{iBr} - K_{iB(r-1)}) \qquad \textit{Equation 18}$$

where T_i is the technique instance, T_{iBR} represents the bandwidth consumption over a set of rounds *R*, and KB_r is the bandwidth consumption in round *r*. This equation is relevant for selecting the best technique from a set of techniques based on their bandwidth consumption. If the change in bandwidth consumption is represented as dK_{Br}, then the change in bandwidth can be defined as follows:

$$T_{iBr} = \sum_{r=0}^{R} dK_{iBr} \qquad \textit{Equation 19}$$

where T_{iBr} represents the technique instance identifier, dK_{iBr} represents the change in the bandwidth as defined by a round, *r* represents the round, and *R* represents the total number of rounds. Typical values for equation 19 for five different techniques were presented and plotted in figure 5.20 back in section 5.7.2.

The bandwidth consumption can also be assessed using the diagram in figure 4.20 to illustrate how the intelligent model operates on this attribute. Thus, this figure is used as a reference in the discussions that follow.

Based on equation 18, the function *f(x)* takes as input each technique's bandwidth consumption value, as defined in equation 19:

$$f(x) = \{ T_i dK_{Br}, T_{i+1} dK_{Br}, T_{i+2} dK_{Br}, T_{i+3} dK_{Br}, T_{i+4} dK_{Br} \} \qquad \textit{Equation 20}$$

where *i* represents the index identifier for each technique, and $T_i dK_{Br}$ represents the bandwidth consumption for technique instance T_i. Just as was done for energy consumption, with equation 13, with respect to bandwidth consumption, the intelligent model's function computes the best technique based on equation 20:

$$f(x) = \{\{O_1, .., O_n\}, \{C_1, .., C_n\}, \{P_1, .., P_n\}, T_{Br}\} \qquad \textit{Equation 21}$$

where, $T_{Br} = \{ T_i dK_{Bir}, T_{i+1} dK_{B(i+1)r}, T_{i+2} dK_{B(i+2)r}, T_{i+3} dK_{B(i+3)r}, T_{i+4} dK_{B(i+4)r} \}$

Computation of Minimal Values

The formula defined in equation 21 defines the intelligent model's function components, which are used to determine the optimal values for the given objective(s) in $\{O_1, .., O_n\}$. Assuming a single technique T_i, which has a set of attributes $\{A_1, ..., A_n\}$, one of which represents bandwidth consumption, and which can hold values constrained by a criterion that defined by the extremes M_{min} and M_{max}. If the main application objective were bandwidth consumption, the function $f(t)$ applies the following formula to determine the best technique from a set based on their bandwidth consumption,

$$f(t) = \{ T_i dK_{Br} < X_{max} \mid T_i dK_{Br} > X_{min} \} \qquad \textit{Equation 22}$$

where Ti represents a technique instance, $T_i dK_{Br}$represents differential bandwidth consumption, and both X_{min} and X_{max} define the constraint extremes for the value of the bandwidth consumption.

5.5.3 Definition of Latency

Latency is defined as the time duration starting from when data is captured by the sensor, up to the point when the aggregated data is submitted to the sink node. Its measurement is in divisions of seconds based on the speed of the network. Latency tends to increase with the number of nodes involved and the size of the network. It represents one of the metrics that can be used to evaluate the performance of a WSN technique.

As discussed for both energy and bandwidth consumption, assuming a WSN network K of n nodes, where N_i represents node I, latency across the network in a round can be calculated using equation 23. This equation is based on the definition of latency to be equivalent to the duration between the moment of data reception on node Ni, to the moment of data reception on node $N_{(i+1)}$, where N_i and $N_{(i+1)}$ are adjacent nodes within the same WSN network, and $N_{(i+1)}$ is already determined, based on the active technique algorithm, to be the next transmission node to N_i :

$$K_L = \sum_{i=1}^{n} N_{(i+1)L} - N_{iL} \qquad \textit{Equation 23}$$

where i is the index identifier for each node instance, K_L represents the network latency in a defined time period, n is the total number of nodes, N_{iL} is the latency of node instance Ni, and $N_{(i+1)\,L}$ is the latency for node $N_{(i+1)}$. defines. The implication of equation 23 includes that both N_i and $N_{(i+1)}$ both have a latency reading. This would be the case only after data transmission from N_i to $N_{(i+1)}$. Usually, this measurement is obtained automatically by the simulation tool via its logging component. Based on the algorithm of the active technique, which is based on the topology (i.e. cluster, tree, chain or mesh), the two nodes, N_i and $N_{(i+1)}$, apart from being in close proximity, must be related by being in the same phase of data transmission towards the sink within the same simulation round, and the data transmission between them is in one direction, and must be from N_i to $N_{(i+1)}$.

Equation 23 could be redefined to associate the latency to each node, by implying that the latency of communication from N_i to $N_{(i+1)}$ is owned by one of the nodes, i.e. N_i, which is responsible for send the data towards the sink node. Based on this rule, equation 23 can be redefined as follows:

$$K_L = \sum_{i=1}^{n} N_{iL} \qquad \textit{Equation 24}$$

where K_L indicates the network latency in the specific time period, i is the index identifier for each node instance, n is the total number of nodes, and N_{iL} is the latency of node instance Ni. This way, latency is directly related to the number of nodes within the network, thereby simplifying the formulas to arrive at differential latencies across the network for different techniques. Equation 24 could also be used to represent the latency across the network in a round by simply associating it with the round r. Re-writing this to indicate the latency for the technique, equation 24 can be re-defined as follows:

$$T_{iLr} = K_{iLr} = \sum_{p=1}^{n} N_{pLr} \qquad \textit{Equation 25}$$

where i represents an index identifier for the technique, T_i represents the technique instance, T_{iLr} represents the techniques latency in round r, K_{iLr} represents latency, under technique T_i, across the network for round r, *while* n is the total number of nodes. p represents the index identifier for each node instance, and N_{pLr} represents the latency for node instance N_p. Based on this relationship, the latency within the WSN for a given technique across the entire length of rounds can be defined based on equation 26:

$$T_{iLR} = \sum_{r=0}^{R} dK_{iBr} \qquad \textit{Equation 26}$$

where i represents the index identifier for the technique, T_i is the technique instance, T_{iLR} represents the latency over a set of rounds R, and dK_{iBr} is the latency, under technique T_i, in round r. This equation is relevant for selecting the best technique based on latency.

Just as with energy and bandwidth consumption, the latency can be assessed based on figure 4.22 to illustrate the behaviour of the intelligent model while taking decisions based on the attribute. To select best technique based on latency, equation 27 is defined:

$$f(x) = \{T_i dK_{Lr}, T_{i+1} dK_{Lr}, T_{i+2} dK_{Lr}, T_{i+3} dK_{Lr}, T_{i+4} dK_{Lr}\} \qquad \textit{Equation 27}$$

where i represents the index identifier for each technique, and $T_i dK_{Lr}$represents the latency for technique instance T_i.

Computation of Minimal Values

Equation 25 indicates in part the set of techniques that would be considered (assuming five techniques involved) when the best technique would be selected based on their latency. As was discussed for energy and bandwidth consumption, the intelligent algorithm will consider the application requirements in order to arrive at a decision. The intelligent algorithm would apply equation 28 to compute the best technique based on this metric.

$$f(t) = \{T_i dK_{Br} < X_{max} \mid T_i dK_{Br} > X_{min}\} \qquad \textit{Equation 28}$$

where i is used as an index identifier for a technique, Ti represents a technique instance, $T_i dK_{Lr}$represents latency for the technique within a round r, and both X_{min} and X_{max} specify application constraints which indicate acceptable limits for the latency.
The next section will combine discussions presented in the section 6.3 and 6.4 to formally analyse the intelligent algorithm.

5.6 Formal Discussion of the function of the Intelligent Algorithm and Machine Learning Model (IAML)

This section presents the formal definition of the intelligent algorithm and machine learning model. The content presented in past sections have covered the analysis of the WSN components, their relationship to WSN attributes, and the classification of attributes. The discussion starts with section starts with the formal analysis of the intelligent algorithm. This is followed by an overview of the machine learning model, where the structure, methods and parameters are discussed. Afterwards, the section concludes with the analysis of the machine learning model behaviour is presented. The input and output attributes discussed in the following major section.

5.6.1 Discussion on the function of the Intelligent Algorithm

This section covers the analysis of the intelligent algorithm relies on equations presented in section 5.4. Equation 14, drawn from section 5.4.1 and shown below, defines the input parameters to the intelligent algorithm, given a round *r* and technique T_i. This equation, though developed for energy consumption, is applicable to all metric attributes.

$$f(x) = \{\{O_1, .., O_n\}, \{C_1, .., C_n\}, \{P_1, .., P_n\}, T_{Er}, K, S\} \quad \textit{Equation 14}$$

where, $T_{Er} = \{T_i dK_{Eir}, T_{i+1} dK_{E(i+1)r}, T_{i+2} dK_{E(i+2)r}, T_{i+3} dK_{E(i+3)r}, T_{i+4} dK_{E(i+4)r}\}$, $\{O_i, ..., O_n\}$ represents the set of objectives, $\{C_1, ..., C_n)$ represents the criteria, $\{P_1, .., P_n\}$ represents the priority list, *K* represents the *Network* entity, and *S* represents the *Scenario* entity. Each $T_i dK_{Eir}$ variable also consists of other attributes as defined earlier in equation as shown below.

$$T_i = \{A_{i1}, A_{i2}, \dots, A_{in}\} \quad \textit{Equation 6}$$

This equation similarly applies to the *Network* and *Scenario* components, as shown below.

$$K_j = \{A_{j1}, A_{i2}, \dots, A_{jn}\} \quad \textit{Equation 7}$$

$$S_k = \{A_{k1}, A_{k2}, \dots, A_{kn}\} \quad \textit{Equation 8}$$

where the relationships between the attributes of the technique, network and scenario was discussed in section 5.3.6. Each attribute is defined using the equation 3 below section 5.3.2.

$$A_m = \{v, T_v, W_v, Th_v, MIN_v, MAX_v, D, S\} \quad \textit{Equation 6}$$

where *m* represents the index identifier for the attribute A_m, *v* represents the value, T_v refers to the type, W_v refers to the weight, and Th_v refers to the threshold. The parameters *D* and *S* represent the attribute's dependence and static/dynamic states respectively. The parameters of MIN_v and MAX_v are applicable when the attribute is used as a metric and provide the opportunity to define constraint limits to the value being held by the attribute. Based on this overview of the essential equations, the algorithm for the intelligent algorithm is discussed below.

Application Requirements

The application requirements determine the rules that the intelligent algorithm applies in its computation. These consists of the objectives, criteria and priority list, as highlighted in equation 14. The components are defined in equations 29, 30 and 31.

$O = \{ A_1, \dots A_i, \dots, A_m\}$ *Equation 29*

$C = \{ A_1, \dots A_j, \dots, A_n\}$ *Equation 30*

$P = \{ A_1, \dots A_k, \dots, A_p\}$ *Equation 31*

Where *i, j and k* is used as an attribute index identifier for all components, *n, m, and p* indicate the number of attributes, *O*, *C*, and *P*, represent the objective(s), criteria, and priority list, and $\{ A_1, \dots A_i, \dots, A_n\}$, represent the attributes that these components consist of.

Objectives: Attributes for objectives cannot be replaced with attributes for criteria and these are incomparable. The attributes under objectives only identify the name of the attribute, for instance, energy consumption. No other attribute is needed to define the objective.

Criteria: The criteria include more detail and needs further attention. Its attributes include the values for the threshold Th_v, the weight W_v, the MIN_v, and the MAX_v. The criteria are defined by equation 32:

$C_{A_i} = \{ Th_v, W_v, MIN_v, MAX_v\}$ *Equation 32*

where *i* is used as an index identifier for the attribute and the corresponding criteria, C_{A_i}represents the *Criteria* defined for a given attribute A_i, Th_v is the threshold defined by the criteria, W_v is the weight, and both MIN_v and MAX_v specify the valid range of values. This definition is used by the intelligent algorithm to determine whether, based on the priority of an attribute, a technique qualifies to be included in a decision phase.

Priority: The priority list in the application requirements consists of the attributes ordered according to importance. The order is defined based on the assigned weighting, W_v. Objectives, criteria, and a priority list will be used as input into an algorithm.

The algorithm referred to above is responsible for combining data generated by simulations, with the set of application objectives, the criteria, and the set of priorities, to select data on right DAT for a given WSN scenario. This algorithm is referred to as an "*intelligent algorithm*" from here on. It is different from the "*intelligent model*", which consists of the machine learning model that is built based on the data created by the *intelligent algorithm*.

Technique Simulation Data

Once the application requirements have been validated, the other data required by the algorithm consists of data generated by the set of techniques. These data will consist of values of the various technique attributes, some of which are static and non-changing, and others dynamic and changing for every round of the simulation. From here on, two techniques, T_1 and T_2 are selected in order to simplify the discussion. Both techniques have their own set of attributes, which will be processed by the intelligent algorithm in order to determine the preferred technique. The computation approach can be applied similarly to a scenario consisting of more than two techniques and the outcome is expected to be the same, in which case, the best technique will be determined. The two techniques are defined as follows:

$$T_1 = \{ A_1, .. A_i, \dots, A_n\} \qquad \textit{Equation 33}$$

$$T_2 = \{ A_1, .. A_i, \dots, A_n\} \qquad \textit{Equation 34}$$

where the attributes, $A_1, .. A_i, \dots, A_n$, for both techniques bear the same attribute name but hold different values. If the attribute A_1 represents energy consumption, it can be defined as follows:

$$A_1 = \{ v, T_v, W_v, Th_v, MIN_v, MAX_v, D, S\} \qquad \textit{Equation 35}$$

where the variables are as defined before. The criteria would have defined values for the parameters W_v, Th_v, MIN_v, MAX_v. The attribute would inherently have a type T_v, which is a real number in this case. It would have values for D, S as well, which are *independent* and *dynamic* in this case. With assumption that the criteria for the WSN application has been defined, to determine the best technique, the intelligent algorithm would perform the following computations. It is essential to note that the attribute parameters consisting of $\{ T_v, W_v, Th_v, MIN_v, MAX_v, D, S\}$ are application-specific, meaning that their values apply across all participating techniques in an application. Only the $\{ v\}$ attribute is different for each technique and is expected to carry a new value at the end of each round. Nonetheless, a use case that would enable multiple definitions for these other parameters would involve more than one application.

Firstly, the intelligent algorithm considers the objective with the highest priority by weight. To simplify the analysis, two objectives are considered in this discussion, energy consumption A_E, and latency A_L, each having weights pre-assigned in the application requirements. This determines their ordering based on the current application. This discussion assumes that energy consumption carries more weight than latency, a result that can be achieved by the intelligent algorithm by using equation 36:

$$f_{priority} = MAX(A_{Ew}, A_{Lw}) \qquad \textit{Equation 36}$$

where $f_{priority}$ implies the function to determine the objective with highest priority, MAX being the mathematical maximum determining function, A_{Ew} and A_{Lw} represent the weights assigned to the energy consumption attribute and latency attributes respectively. As noted earlier, these apply to all techniques. Assume that this equation places A_{Ew} before A_{Lw}, that is energy consumption before latency. Then the intelligent algorithm will need to determine which techniques fall into the range defined by the criteria parameters MIN_v , MAX_v. This computation is applied to the attribute values for both techniques (and to others if there are more). The following equations are used by the algorithm:

$$f_{range} = \{x: x > MIN_v \ and \ x < MAX_v\} \qquad \textit{Equation 37}$$

where f_{range} specifies the range validation function, x is the value of the attribute, that is any of energy consumption and latency in this case, and MIN_v, MAX_v are the criteria defined value constraints for the attribute. The equation is applied to the relevant attributes of all participating techniques. Equation 38 shows this being applied to technique T_1*'s* energy consumption attribute value:

$$f_{range} = \{A_{Ev_{T1}}: A_{Ev_{T1}} > MIN_{Ev} \ and \ A_{Ev_{T1}} < MAX_{Ev}\} \qquad \textit{Equation 38}$$

where f_{range} represents the range validation function, $A_{Ev_{T1}}$ represents the energy consumption value for technique T_1, after a specific round *r* (not identified in this equation for simplicity), MIN_{Ev} and MAX_{Ev} both specify the criteria range. Equation 36 is also applied to the energy consumption attribute of technique T_1, as well as the latency attributes of both techniques. The outcome of the function includes a collection of the techniques whose attributes satisfy the condition of the equation.

The final stage of the intelligent algorithm's function involves comparing attribute values for all participating techniques. This is based on the form of the objective. For instance, energy consumption would involve a minimisation function, while network lifetime would involve a maximisation function. Continuing with the energy consumption attribute selected earlier, equation 37 would be used by the algorithm to determine the best technique based on the current round and application:

$$f(T_{optimal}) = MIN(A_{Ev_{T1}}, A_{Ev_{T2}}) \qquad \textit{Equation 39}$$

where $f(T_{best})$ defines the function to select the best (optimal) technique given the conditions, MIN is the mathematical minimum function, and $A_{Ev_{T1}}, A_{Ev_{T2}}$ represent the energy consumption values for techniques T_1 and T_2. The outcome of this equation is the selected best technique to be applied in the subsequent simulation round.

Combination of Application Requirements and Technique Simulation Data

The last two sections present an analysis of the intelligent algorithm's functions and describes the approach used to select the best technique given the WSN context. The attributes that contribute to concluding on this decision are discussed in detail in section 5.5.3. Only energy consumption and latency are discussed. The same discussion applies to bandwidth consumption and other such attributes. However, other attributes, not mentioned, such as bandwidth consumption, are valid for inclusion as well. The equations above apply to all WSN techniques and attributes, beyond those mentioned. For the intelligent algorithm to perform this function, the technique attribute data is needed for all participating techniques based on the same WSN conditions. Figure 5.4 illustrates the data exchanged in the interface between the intelligent algorithm and the machine learning model. These include a scenario S_1, network N_1, the application requirements, and the best selected technique T_1 based on this combination.

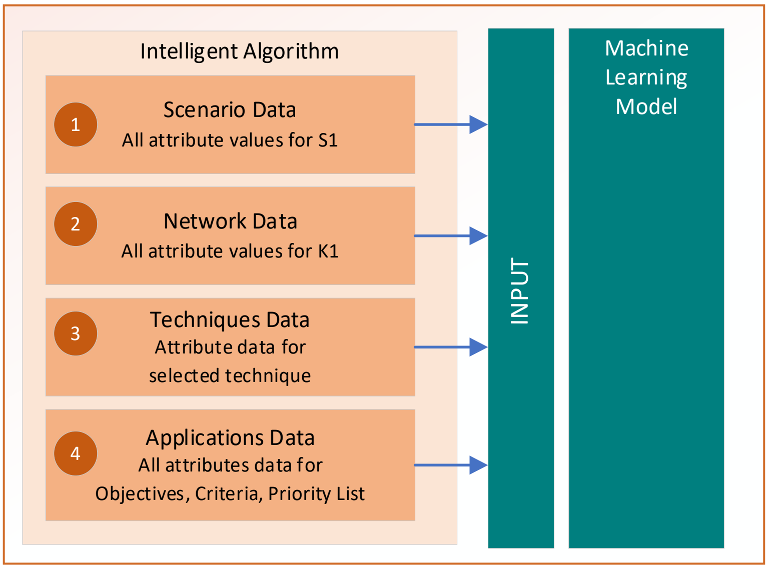

Figure 5-4 - Interface between the Intelligent Algorithm and the Machine Learning Model

The attributes identified in figure 5.4 form the input vector for the machine learning model. The details of these input attributes are discussed further in section 5.5.3, where the number of input nodes (or units) to hold the data is computed for the machine learning model.

5.6.2 Discussion on the Machine Learning Model

This section presents design details of the machine learning model and follows on with the formal definition of its operation. Once the intelligent algorithm has completed the data processing, the next stage in the process involves the machine learning model (ML model). The output data from the intelligent algorithm is used to train the ML model, which subsequently recognises the patterns in the data. Once trained, the model is used to make predictions and recommendations based new application scenarios.

The ML model is expected to discover patterns representing relationships between the three components consisting of (1) the provided application requirements, (2) the WSN settings, represented by the *Network* and *Scenario* entities, and (3) preferred WSN data aggregation technique based on the combination (represented by the *Technique* entity). The output of the ML Model provides the relative probability that a specific technique is the best option for the given application requirements. The input data to the ML model is summarised in table 5.2. Since the pattern needs to be discovered and learned, a multi-layer artificial neural network (ANN) had been preselected as the preferred machine learning model for this purpose.

Table 5-2 - Input data to the ANN, which is provided in the output of the Intelligent Algorithm

No	Data	Description
1.	Application Requirements	This consists of the application or scenario objectives, criteria (or constraints), and priority list
2.	Network data	This consists of network related data, such as the number of nodes, their location and distribution.
3.	Scenario data	This consists of scenario or event details, such as event type, sampling rate, etc.
4.	Technique Data	This consists of the attributes of the technique considered appropriate for the given scenario.

The ANN consists of various layers, which consist of the input, multiple hidden layers, and the output layer. The shape of its input layer is discussed further in this chapter. Its output is defined by the number of techniques considered in the training session. Further details on the layers are also discussed in following sections.

Description of the function of the Artificial Neural Network

Figure 5.5 presents a high-level structure of the ML model and serves as a reference point for the following discussions. The configuration of each layer is discussed in more detail in subsequent sections. The input layer is labelled *I*, the hidden layers H_1, H_2, H_3, the SoftMax layer *S*, and output layer *O*. Three layers were pre-selected for this model based on common standards used in similar experiments found in literature. The *SoftMax* layer, applies the SoftMax activation function, being a multi-class predictive model.

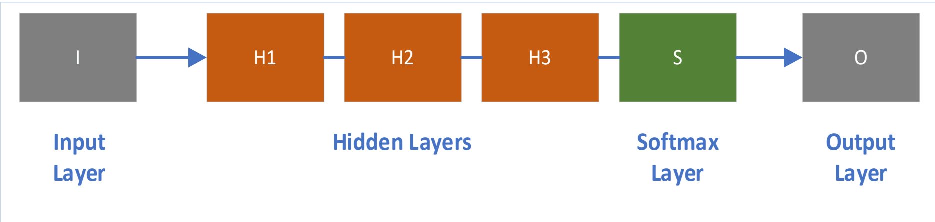

Figure 5-5 - Structure of the Artificial Neural Network (ANN)

The number of units in the input layer is equivalent to the shape vector of the input data to be used for training. The number of units in the hidden layers are modified until exact numbers are obtained to achieve optimal output from the model. While the hidden layers apply the ReLU activation function, the SoftMax layer provides a probability value for each of the techniques. However, ahead of discussing the internal details of the ML model, the formal discussion on the final stage of the intelligent algorithm/machine learning model system (interchangeably referred to as *Intelligent Model* from here on) is presented below.

ML Model Selection for Best Technique (Final step of process)

The final step of the operation involves the training of the ML model, which enables it to select and recommend the best technique given the set of requirements and WSN context. In discussions presented in prior sections, two techniques were selected for simplicity. Here, the two techniques are assumed to be the available techniques in the simulation and only one of these can be the best technique. Equation 38 defines the function of the ML model and highlights the variables it accepts as input from the intelligent algorithm to perform the technique selection function.

$$f_{input}(x) = \{\, O_A, C_A, P_A, T_{best}\} \qquad \textit{Equation 38}$$

where $f_{input}(x)$ implies input function to the ML model, O_A refers to the set of objective attributes, C_A refers to the set of attributes that form the criteria, P_A refers to the priority list, defined by the weights assigned to the attributes, and T_{best} identifies the best technique selected by the intelligent algorithm. It is noteworthy that the two techniques, T_{1_A} and T_{2_A} , were candidate techniques in this scenario and only one of them was selected to become T_{best}. Equation 38 represents the high-level structure of the input vector that serves as input into the ML model. A separate query channel to the ML model for prediction or recommendation results is defined in equation 39. The vector as illustrated here lacks data on any specific technique, since this information would be provided by the ML Model.

$$f_{query}(x) = \{\, O_A, C_A, P_A\} \qquad \textit{Equation 39}$$

where $f_{query}(x)$ represents the query input function to the ML model, O_A refers to the set of objective attributes, C_A represents the set of criteria attributes, and P_A represents the priority list. The ML model provides output defined by equation 40, and this represents the proposed recommendation for the best technique provided by the ML model.

$$f\left(T_{output}\right) = \{T_{1_{p1}}, \dots, T_{i_{pi}}, \dots, T_{n_{pn}}\} \qquad \textit{Equation 40}$$

where $f\left(T_{output}\right)$ represents the output vector of the ML model, $T_{i_{wi}}$ represents the probability assigned to the *ith* technique, while *n* represents the number of candidate techniques in the simulation. The component parts of T_{i_p} are defined in equation 41.

$$T_{i_p} = \{T_i, p\,\} \qquad \textit{Equation 41}$$

where T_i represents the technique, and p represents the probability assigned to the technique based on the input vector data. The ML model's intricate learning process takes place between equations 38 and 40. However, by the time equation 40 is performed, the ML model should have provided a recommendation, based on its learning rate, the best recommended technique, application requirements, WSN context, and the participating techniques.

The next section presents the details of the machine learning model, its methods and activation function. The following section discusses the format of the input and output attributes and provides detail on the transformations necessary to pre-process the raw data for ML model.

5.7 Further Description on the Reference Architecture

The discussion in this section continues from section 5.2, and convers the reference architecture from an implementation viewpoint. Section 5.2 discussed the design of the process for building the intelligent algorithm and machine learning model. This section provides details on the relationship between those discussions and highlights how the equations discussed so far relate to the architecture. This is represented in figure 5.6. The equations are shown in circles with one of two labels: *Ex*, for equations, where x implies the equation number, and *Rx*, for relations, where x represents the relation number.

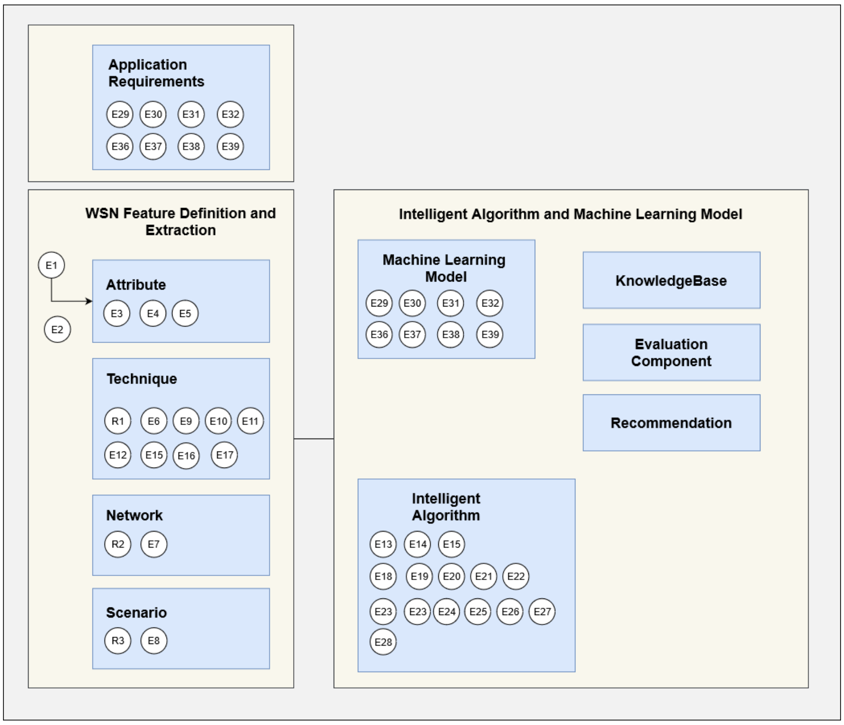

Figure 5-6 – Placement of equations in corresponding components in the intelligent model.

5.7.1 Discussion of Input and Output Attributes

This details of the input and output data for the intelligent algorithm and machine learning model are discussed in this section. The deliverables from this discussion is used to determine the attributes that will provide data to the intelligent algorithm, and after encoding, the number of input nodes to the ML model. Once the number of the input nodes are determined, the ML model's input vector shape can be well-defined.

Definition of Input and Output Features

Some of the WSN attributes discussed in this section were defined earlier in section 2.5, and further in table 2.2. Many of the attributes serve as input parameters for the intelligent model. The guideline discussed in prior sections is used here to categorise the attributes as falling under one of the WSN entities.

Figure 5.7 provides an overview of encoding methods used in pre-processing of the input data. The data types are categorised under classes (A, B and C) as illustrated in the figure.

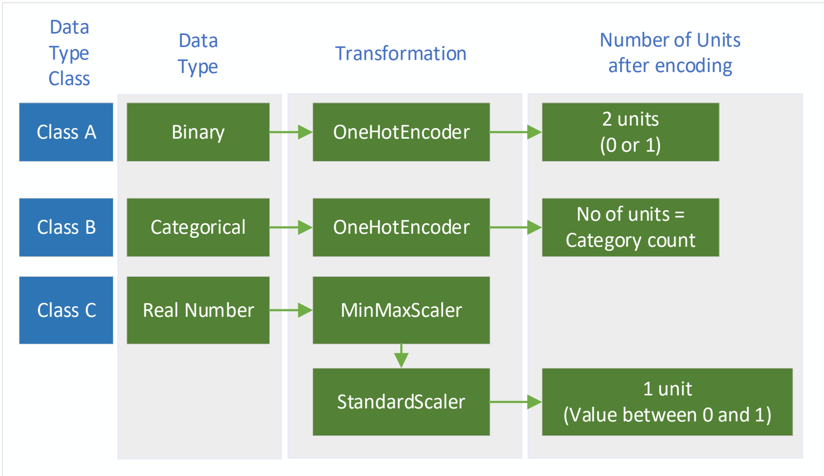

Figure 5-7 - Illustration of the feature types, their required encoding, and the equivalent input node requirements

5.8 Identification of Data Sources

In this section, final set of attributes, and the data sources used for data collection are discussed. The definition of the attributes relates back to discussions held in sections 2.7 and 4.6 but includes the introduction of new (derived) attributes. These are used later in determining the type of training data required for the intelligent model. The following primary sources were selected for collection of data:

- Articles and journals
- Expert sources
- Simulated data generated based on settings obtained from relevant expert, real or experiment sources
- Simulated data generated strictly within simulation

The above sources are further explained in table 5.3. While the data collected from these sources could be categorised into those collected from realistic scenarios and those generated from simulation, others could be classified into simulation-generated but based on real-scenario settings.

Table 5-3 - Data Source Categorization

Data	Purpose
Articles and Journals	These consist of published articles and journals that have investigated various WSN data aggregation use cases. Such experiments identify relevant WSN attributes and metrics that can capture and evaluate the WSN's performance. Some of the attributes selected from such sources are presented in table 6.8. For example, they include the attributes such as energy consumption, bandwidth, throughput, and latency.
Expert sources	Expert sources include practical applications of WSNs in real-life scenarios. Such sources identify attributes that are relevant for monitoring and evaluating real-life events and some of these attributes are also shown in table 6.8. Some of such sources include the United States Geological Survey (USGS), British Geological Survey (BGS), and European-Mediterranean Seismological Centre (EMSC). For example, physical node distribution based on the scenario could dictate the appropriate topology in many use cases, as well as the sampling rate.
Simulated data, based on expert-driven settings	These consist of data generated from simulation with environment settings dictated by either article, journal or expert settings. For instance, WSN experimental setups usually rely on a common power supply model, radio configuration, range of node energy levels, and radio transmission rates.
Strictly Simulated data	This consist of data that can only be obtained from the simulation environment. These include specific network attributes such as network coverage, latency, energy consumption, etc.

Figure 5.8 presents a high-level architecture of the interaction between data sources and other components of the system, which includes the collected data, the intelligent model and model evaluation components. The squared numbers are used to enable reference to various components. The boxes numbered 1, 2, 3, and 4, identify the four data sources. The extracted data from these sources are indicated in the second column titled "*Data*", which include data on the scenario, the application/scenario requirements, the network, and technique attributes. The third column titled the "*Intelligent Model*", identifies the intelligent model, while the fourth column named "*Evaluation*" identifies the subsequent evaluation and testing of the model after training. The box numbered 5 identifies benchmark data and provides data that is used later for model evaluation. Tasks such as data pre-processing are not indicated in this figure but are discussed later following the specification of attribute fields.

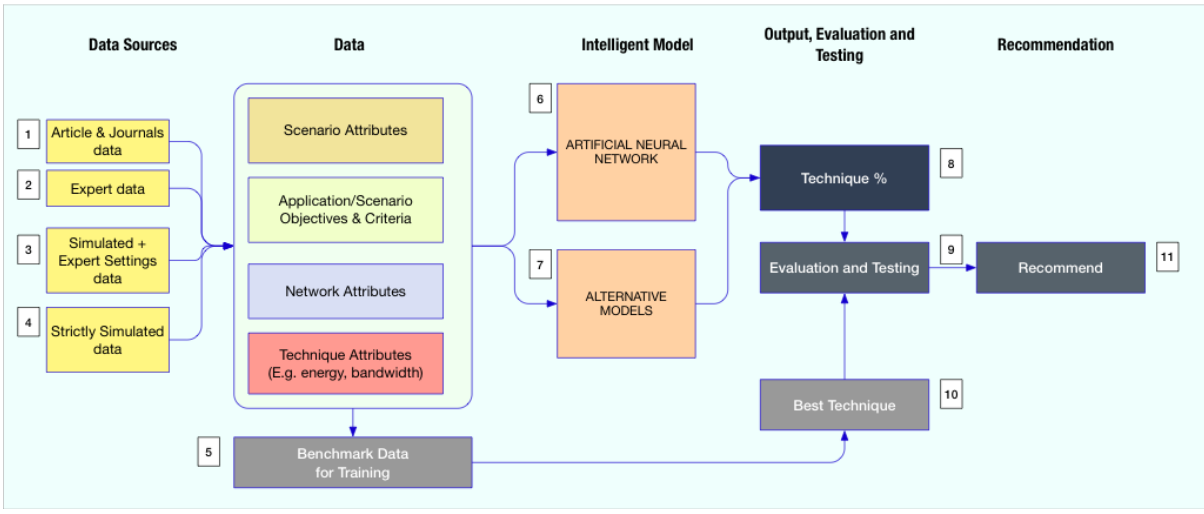

Figure 5-8 - High-level Intelligent Model Architecture – shows the identified data sources, the type of data obtained from the sources, the intelligent model, evaluation and testing and recommendation.

The selection of many features used for the intelligent algorithm and machine learning model was based, or directly related to, similar attribute selections made in research focused on the optimization of WSNs. In the past several years, machine learning had been explored as a method to optimise WSN performance in various scenarios and many of these studies have been referenced in this study (Hooda et al., 2018; K. and Vaidehi, 2018; Praveen Kumar et al., 2019). The relevant attributes considered in this study are discussed in table 5.4. the attributes are categorised based on their type (numeric, binary, or categorical), and the categorical values (if data type is categorical).

Table 5-4 - Data source attributes identified for the intelligent model. Some of the data are collected from experiments and real scenarios, while others are generated from simulation, in a few cases, based on settings collected from real scenarios. Only the relevant features are used to train the ML model.

S.No.	Attribute	Data Type	Category Values	Comments
1.	EnergyObjective	Numeric	Not Applicable	Objectives are passed as real numbers on separate inputs with their values determining their priority.
2.	BandwidthObjective	Numeric	Not Applicable	
3.	LatencyObjective	Numeric	Not Applicable	
4.	ConnectionsInRound	Numeric	Not Applicable	Computed from simulation
5.	AvgDistanceToSink	Numeric	Not Applicable	Computed from simulation
6.	ShortestPathToSink	Numeric	Not Applicable	Computed from simulation
7.	LongestPathToSink	Numeric	Not Applicable	Computed from simulation
8.	MinEnergyConsumption	Numeric	Not Applicable	Criteria values are either submitted manually, or autonomously detected based on the scenario characteristics
9.	MaxEnergyConsumption	Numeric	Not Applicable	
10.	MinBandwidthConsumption	Numeric	Not Applicable	
11.	MaxBandwidthConsumption	Numeric	Not Applicable	
12.	MinLatency	Numeric	Not Applicable	
13.	MaxLatency	Numeric	Not Applicable	
14.	EnergyConsumption	Numeric	Not Applicable	Obtained from simulation
15.	Latency	Numeric	Not Applicable	Obtained from simulation
16.	BandwidthConsumption	Numeric	Not Applicable	Obtained from simulation
17.	NetworkConnectivityRadius	Numeric	Not Applicable	Value determines the radius used to compute various parameters: *NetworkCoverage*, *NetworkConnectivity*, etc.
18.	NetworkCoverage	Numeric	Not Applicable	Computed from simulation. Should reflect target coverage of scenario
19.	NetworkConnectivity	Numeric	Not Applicable	Metric based on node-to-node connectivity in the in a round
20.	SamplingRate	Numeric	Not Applicable	Captured from scenario or assumed for simulation environment
21.	CommunicationAlgorithm	Categorical	Hierarchical, Flooding, Diffusion	Determined from scenario

22.	FieldSize	Numeric	Not Applicable	Data is available from either the real-life scenario or the simulation environment. In recommendation state, it would be required as an input.
23.	HomogenousNodes	Binary	True, False	
24.	AggregationNode	Categorical	Sink, Intermediary, Heads	
25.	SensingTrigger	Categorical	Query, Event, Continuous	Determined from scenario
26.	PeriodicBasedReporting	Numeric	Not Applicable	These values are based on the value held by AggregationType
27.	EventBasedReporting	Numeric	Not Applicable	
28.	RealtimeMonitoring	Numeric	Not Applicable	This field is represented as real since a scenario could involve a fraction of both
29.	QueryMonitoring	Numeric	Not Applicable	
30.	PysicalSensorTopology	Categorical	Star, Bus, Linear, Ring, Mesh	Determined from scenario
31.	RateOfSpread	Numeric	Not Applicable	Determined from scenario
32.	PrimaryMedium	Categorical	Solid, Liquid, Gas	Determined from Scenario
33.	TransmissionRange	Numeric	Not Applicable	Computed
34.	InitialNodeEnergy	Numeric	Not Applicable	Computed
35.	LocationAwareness	Binary	True, False	Determined from scenario
36.	SinkReportingMode	Categorical	One-to-One One-to-Many Many-to-One Many-to-Many	Determined from scenario
37.	NodeMobility	Binary	True, False	N/A
38.	NodeDistributionRatio	Numeric	Not Applicable	Metric based on nodes per square meter^2
39.	FractionOfHeads	Numeric	Not Applicable	N/A
40.	NumberOfPackets	Numeric	Not Applicable	N/A
41.	PacketSize	Numeric	Not Applicable	Computed from simulation
42.	NumberOfNodes	Numeric	Not Applicable	Computed from simulation
43.	ActiveNodesInRound	Numeric	Not Applicable	Computed from simulation
44.	TotalSentPackets	Numeric	Not Applicable	Computed from simulation
45.	MinNextNodeDistance	Numeric	Not Applicable	Computed from simulation
46.	MaxNextNodeDistance	Numeric	Not Applicable	Computed from simulation
47.	AvgNode2SinkDistance	Numeric	Not Applicable	Computed from simulation

48.	AvgNextNodeDistance	Numeric	Not Applicable	Computed from simulation
49.	AvgHeadsCoverage	Numeric	Not Applicable	Computed from simulation
50.	HeadsToNodesRatio	Numeric	Not Applicable	Computed from simulation
51.	BestTechnique	Categorical	Leach, Heed, Pegasis, Dbst	Best technique is provided by expert scenario

Table 5.5 provides an example of typical settings assigned to some attributes in a typical WSN experiment.

Table 5-5 - Example WSN experiment showing sample values that are assigned to specific attributes. With respect to these set of attributes, other attributes such as energy consumption, are considered dynamic attributes, which change based on a behaviour defined by the values assigned to the attributes in this table (including others not mentioned here). The last column in the table explains how the attributes behave in an event or simulation.

Attribute	Typical Experimental Value	Conditions for Change
Number of initial nodes	100 nodes	The initial number of nodes when the event of simulation starts. This number could reduce as the event proceeds as nodes start to exhaust their energy level. At that point the number of active nodes will become relevant.
Field size	50 m x 50 m (meters^2)	This value could change as the event proceeds if such event is considered spatially dynamic, such as a wildfire.
Initial Node Energy	2J or 50J	This changes as nodes transmit in each round. The sum of this value for every node in a round provides the network energy
Sampling rate	600kbps	Realistically, this could change based mainly on the states of monitoring, detection and tracking. The scenario or event type, such as earthquake, or wildfire, also impacts on this value.
Base station location	X = -25m, Y = -25m	This is usually relative to a reference point of (0m, 0m), and usually remains static throughout an event.
Energy consumption per bit	50 nJ/bit (nano-Joules per bit)	This also remains static throughout an event and hardly changes unless there is a change to the node hardware
Packet size	6kb (kilobytes)	This remains static throughout an event
Network density	0.01 node/m^2	This is determined dynamically by the number of active nodes within a given field size.

Table 5.6 and table 5.7 both further illustrate how the attributes can be combined to achieve the goals of this research, by highlighting the possibility of training of the ML Model. Table

5.6 shows the attributes for a selected technique, where Leach is chosen in this case. Table 5.7 shows typical data for a selected event; a forest fire is chosen in this case. The attributes shown in table 5.7 indicates typical event characteristics, highlighted from expert sources, which could affect the behaviour of the appropriate event. The attributes, which include latency and location awareness, could be considered inherent characteristics of the event, while the values of the attributes could be considered as inherent requirements for the event. The need to monitor the event defines the need for a minimum level of latency, while also requiring an awareness of the location of the event, or the sensor nodes themselves. Without indicating that the Leach technique is considered the best technique for data aggregation in this case, it becomes obvious that the same attributes can be compared with those of various techniques in order to select the best technique (considering various other attributes, or parameters).

Table 5-6 - Sample attribute values for a WSN technique. The Leach technique is used in this case.

Technique	Attribute	Value
Leach	Primary Objective(s)	Energy Consumption, Network Lifetime
	Location Awareness	False
	Energy	250J (All nodes)
	Latency	0.05 seconds

Table 5-7 - Sample scenario or event data indicating possible values for a forest fire event

Event	Attributes	Setting or Requirement
Forest Fire	Primary Objective(s)	Network Lifetime, Latency
	Location Awareness	True
	Latency (Minimum)	10 mins

In the next few pages, analysis of typical WSN use case data is presented. Table 5.8 provides some of the captured data for various WSN events and scenarios. It represents an extended version of table 5.7, which was discussed above. It contains the data captured for various scenarios, across various attributes, composed mainly from multiple research sources. The attributes are used to specify or model various characteristics of the events. Most of the events are classified under monitoring and detection, since these two states have a huge impact on some characteristics, such as sampling rate and active node count (some nodes are kept off during monitoring for instance).

Table 5-8 – Various WSN use case data collected from multiple sources including real-life and experiments. Some of the data can only be determined in simulation since they are considered dynamic. The tick indicates that the feature is relevant for the specific event.

	Events						
Features	ForestFire Monitoring	ForestFire Detection	OilGasPipeline (Surface) Monitoring	OilGasPipeline (UnderWater) Monitoring	AirQuality (CO, CO2, NO2, O3, H2S) Monitoring	Earthquake Monitoring	Earthquake Detection
Objectives							
Maximize Network Lifetime	✓	✓	✓	✓	✓	✓	✓
Minimize Latency	✕	✓	✕	✕	✕	✕	✕
Maximize Accuracy	✕	✕	✕	✕	✕	✓	✓
Minimize Energy	✕	✓	✕	✕	✕	✕	✕
Minimize Bandwidth	✕	✓	✕	✕	✕	✕	✕
Sampling Rate (SR)							
SR1 - Very Low (x > 1 min)	✓	✕	✕	✕	✕	✓	✓
SR2 - Low (x > 10 sec)	✕	✕	✕	✕	✕	✕	✕
SR3 - Medium (1 sec < x < 10 sec)	✕	✓	✕	✕	✕	✕	✓
SR4 - High (x < 1 sec)	✕	✕	✕	✕	✕	✕	✕
SR5 - Very High (x < 1 msec)	✕	✕	✕	✓	✕	✓	✕
Field Size (FS)							
FS1 - (10m < x < 30m)	✕	✕	✕	✕	✕	✕	✕

FS2 - (30m < x < 50m)	✕	✕	✓	✓	✕	✕	✓
FS3 - (50m < x < 100m)	✕	✓	✕	✕	✕	✓	✕
FS4 - (100m < x < 150m)	✓	✕	✕	✕	✕	✕	✕

The foregoing discussions have provided a design and reference guide for the next stage of the research, which involves the data collection process, data pre-processing, machine learning model development, training, and evaluation. The next section covers the development of the machine learning model.

5.9 Justification for Intelligent Model Framework

As highlighted earlier, machine learning is used in this study to build the model to intelligently determine a data aggregation technique given event conditions. This choice is essentially driven by the need to process a high volume of data to detect patterns in current WSN scenarios. Also, via various experiments, the unpredictable nature of the environments into which WSNs are deployed, and their subsequent behaviour, cannot be determined using ordered or mathematical means (Bangotra et al., 2018; Praveen Kumar et al., 2019).

The use of machine learning in WSNs has included the use of algorithms within the three main categories, i.e. supervised, unsupervised and reinforcement learning. Algorithms that have been used include Support Vector Machines (SVMs), Bayesian statistics, Decision Trees, Neural Networks, and K-Nearest Neighbour (supervised), k-means clustering, Principal Component Analysis (Unsupervised), and Q-learning technique (reinforcement learning), Restricted Boltzmann Machine (either supervised or unsupervised), (Bangotra et al., 2018; K. and Vaidehi, 2018; Khan and Samad, 2017; Otoum et al., 2019; Praveen Kumar et al., 2019). Various machine learning algorithms have been used for specific WSN scenarios, while research has also been done to assess their suitability to different WSN use cases (Kumar Dwivedi et al., 2018; Praveen Kumar et al., 2019). As stated earlier, this research will apply a multi-layer artificial neural network (ANN) for the target machine learning model. The details of the model are discussed further in subsequent sections.

The next few sections of this chapter cover more practical aspects of the study. These include the simulation environment design, attribute details, class relationships, implementation details, and benchmark specification.

5.10 Simulation Environment Description

This section provides some details about the simulation environment, and default settings used during the simulations.

5.10.1 Details of Simulation Environment

The simulation environment was based on the version 3.30 of the NS3 simulator, i.e. Network Simulator 3 (Nsnam, 2020). NS3 is a discrete-event network simulator targeted mainly at research and development in wired and wireless network simulations. It is an open-source tool and is actively being supported by a collection of organizations grouped under the University of Washington NS-3 Consortium. The tool provides an environment that facilitates simulation, configuration, execution, trace collection and analysis of a network of nodes. Being developed in the C++ language, it provides facilitates the development of complex models and protocols for experimentation in wireless sensor network simulations.

5.10.2 Software Environmental Settings

In order to perform the simulations under the NS3 tool, the environment was setup according to the following details:

- Hosting Environment: Oracle VirtualBox/Linux, a virtual environment, was created to install a Linux-based operating system (Ubuntu), which was the suggested environment for using NS3. The installation platform had a RAM of 32GB, and a hard drive space of over 40GB.
- Development Environment: the development environment consisted of a combination of PyCharm and Eclipse. PyCharm enabled use of Python to perform data pre-processing and building the intelligent model. Eclipse enabled development of complex models within the NS3 tool using C++.

In their default settings, the above tools were used to achieve the task of developing and running the simulations and training the intelligent model.

5.11 Metrics

This section discusses selected performance metrics that are used to evaluate the techniques in specific WSN scenarios. Two groups are defined: *Standard metrics*, which inherently represent the performance of techniques in WSNs, and are calculated from primary attributes; and *Non-standard metrics*, which are relevant when considered with standard attributes, and are derived from a collection of attributes. The non-standard metrics are developed based on the outcome and learning from experiments performed on the techniques.

5.11.1 Standard Performance Metrics

A few specific wireless sensor network attributes were selected to study the comparative behaviour of the different techniques. This information provided insight into the performance of the WSN during the technique modelling and experimentation and a foundation for further comparative analysis in given scenarios. While the values of independent variables are affected directly by changes in the network, their values contribute to generating the values for the metrics. Thus, the metrics can be used to evaluate the behaviour of the technique. Table 5.9 presents details of the selected metrics. The details discussed in table 5.9 was carried out in section 4.8.

Table 5-9 – Selected metrics used to evaluate the performance of techniques. They are referred to as Standard metrics and computed by reading a single attribute during simulation.

No	Standard Metric Name	Description	Independent Attributes (based on a single independent attribute)
1.	Energy Consumption	WSN nodes transmit and receive data, and by this activity, consume energy. Given the active technique, different energy consumption trends were recognised and was used to evaluate the performance of different techniques.	Node Energy (Joules)
2.	Bandwidth Consumption	Bandwidth is consumed by the nodes when they transmit across the network. Based on the technique, different consumption trends were recognised, and this was plotted for various techniques. this characteristic was also used to evaluate the performance of the techniques.	Node bandwidth (bytes)
3.	Latency	The duration of packet transfer between two nodes can be referred to as the latency of the communication. Summation of this across the network was also used as a yardstick to evaluate the performance of various techniques.	Packet delivery and Network-wide Latency (milliseconds or nanoseconds)

5.11.2 Derived Performance Metrics

A few other derived attributes were defined and are described in table 5.10. These are dependent variables and computed based on a set of independent variables. They provided a computational metric to facilitate the evaluation of techniques during simulation as well as during the model development. However, these are not featured in the training data for the intelligent model. Their values were captured throughout the simulation to provide guidance towards understanding the relationship between various standard attributes such as how the standard attributes relate to the field size. Here, they are referred to as *Non-standard attributes* since they do not measure the primary characteristics of the network. They are listed in table 5.10.

Table 5-10 – Additional sub-metrics used to evaluate the performance of techniques – referred to as non-standard metrics, and computed based on a collection of standard attributes.

Non-Standard Metric Name	**Description**	**Dependent Attributes** (resulting attributes based on combining independent attributes)
Distribution Factor	This represents an attribute that was planned to model the distribution of nodes. The location of nodes with respect to other nodes directly affects their transmission load, and thus, their energy consumption. It was calculated by taking into consideration the location of a node, either with respect to the location of the sink node, or to the set of adjacent nodes based within a radius of the node. This value is not expected to change throughout a single simulation of many rounds.	• Average node distance to sink node or head node (metres) • Node density – number of nodes in a radius around the node (constant) • Number of heads within node radius (nodes/metres^2) • Distance between closest head and sink node(metres)
Proximity Factor	This defines the distance between a node and its next transmission node. This attribute was used to create a proximity factor for each node, which affected its energy consumption.	• Average distance between a node and its target transmission node (metres) • Average node distance to all nodes within a radius (metres)

Communication Factor	This network metric was used to represent the overall communications capability of the network of nodes. It is expected to reduce over the lifetime of the WSN application. It was computed by taking into consideration the combined transmission range/energy of all nodes, the distance between the closest two nodes, the path distance from each node to its head, and the distance of the nearest head to the sink.	• Distance between a node and its next node (metres) • Required energy for transmission between a node and its next node (Joules) • Distance between a node and either the sink or head node (metres)
Coverage	This metric was used to model the effective area covered by a sensor based on a given radius. It was computed by taking into consideration the transmission range of each node, and alternatively by considering a radius around the node, including the proximity of the closest node to a primary node.	• Node transmission radius (metres) Node coverage based on a given radius (nodes/metres^2) • Combined node perimeter between two or more nodes (metres) • Node coverage based on given radius of network (nodes/metres^2)

5.12 Summary

This chapter covered various topics concerning the design and development of the intelligent model. It discussed extensively the mathematical model for the behaviour of the entire system, covering how the WSN entities, their attributes, and metrics, how these relate to the application requirements, and how they are integrated to facilitate the intelligent model. Then it discusses the data design, the system interfaces, and the system overview, where system integration with external components are discussed. It concludes with details of the simulation environment, as well as various additional attributes that were computed during simulation to help with understanding the data. However, its testing and implementation are carried out in subsequent chapters.

The next chapter discusses the intelligent model and its prototype implementation design.

6 Implementation of the Intelligent Model and Software Prototype

6.1 Overview

This chapter discusses the implementation of the intelligent model, as well as the software prototype. The last chapter discussed the analysis and design of the intelligent model, where various supporting concepts were introduced. These concepts are used in this chapter to develop an implementation of the model. The software prototype consists of a client-facing application, which includes the source code developed within the simulation environment, the data generated, as well as a client interface.

Relevant hardware and software tools are also discussed. The software prototype, as well as how it is applied I towards evaluating the intelligent model is discussed. A process design for using the software prototype is also discussed. From here on, the chapter refers to parameters in the term “hyper-parameters” as meaning the parameters used to configure the machine learning model.

6.2 Hardware and Software Tools

This section covers details of the hardware and software used in the development of the intelligent model and the prototype used in evaluation.

6.2.1 Hardware Description

The hardware details used in the development of the system are stated in table 6.1. The development and simulation environment involved a Windows 10 64bit platform running an Oracle VirtualBox virtual machine. The virtual machine provided the environment to install and run the NS3 network simulation software, which was used to simulate the wireless sensor network scenarios.

Table 6-1 - Details of the hardware used in development of the intelligent model and prototype

Specification	Detail
Intelligent Model & Prototype	
Platform/Operating System	Windows 10, 64-bit (x64)
Processor	Intel Core i5-6440HQ CPU @ 2.60 GHz 2.59 GHz
RAM	32.0 GB
Hard Disk Drive	250 GB (70 GB Free)

Web Server	Linux 4.16.10-300.fc28.x86_64 x86_64
Simulation Software	
Platform/Operating System	Windows 10 – 64-bit (x64) / Ubuntu 18.4 LTS (Virtual Machine in Oracle VirtualBox)
Processor	Intel Core i5-6440HQ CPU @ 2.60 GHz 2.59 GHz
RAM	32.0 GB
Hard Disk Drive	250 GB (70 GB Free)

6.2.2 Software Tools Description

The details of software tools used for the development of various software components of the system are described briefly in table 6.2, categorised according to the software component. The flexibility of simulation environment, i.e., NS3, enabled the development of WSN models and automatic parsing of output data based on specific formats, which required minimal pre-processing before ML model training.

Table 6-2 - Details of the software used in development of the intelligent model and software prototype

Specification	Detail
Intelligent Model & Prototype	
Development Environment	PyCharm 2020.2.3 (Community Edition)
Programming Language	Python 3.6
Software Frameworks/Modules	TensorFlow 2.3.1, Keras 2.4.3, NumPy 1.19.4, Pandas 1.14, Matplotlib 3.3.2, CSV, Scikit-learn 0.23.2
Programming Language, Scripting	HTML, CSS, JavaScript, Python Django
Simulation Software	
Simulation Environment	Network Simulator (NS3)
Simulator Version	NS3.29 (ns-allinone-3.29)
Programming Language	C++ (gcc), Python
Development Environment (IDE)	Eclipse, Visual Studio Code
Frameworks/Namespaces	C++ standard libraries, ns3 libraries (NS3)

The software prototype stands as a separate component of the system and only queries the intelligent model for recommendations based on single sample use cases. Though useful, its evaluation was already captured in the various evaluations that are captured in the following chapter. Nonetheless, it's design and development were included in order to provide an initial stage to future integrations to the intelligent model via REST APIs for instance. It was designed to be hosted on a web server, which was based on the Python Django framework. Based on this implementation, an interface could be created for a human user to directly query the intelligent model for its recommendation. However, it is important to note that the typical scenario that would consume the services of the model would be more of machine-to-machine (M2M) communications.

6.2.3 Development Challenges and Constraints

With regards to the software details presented in table 6.2, a few constraints were encountered during the simulation and data gathering process and this are discussed below:

- NS3 Simulation Environment:
  - Installation Environment: the prescribed installation platform for the NS3 simulation tool as a Linux environment. Though, a few other equivalent simulation tools were discovered, NS3 was chosen for its single language platform and ease of data manipulation within the same environment. It was also a preferred simulation tool in wireless sensor networks research.
  - Programming Language: the programming language of the tool is C++. Its predecessor, NS2, included two different developing models, one based on C++, and the other a scripting environment, enabled automation of certain parts of model development. NS3, being based on C++, required some of such components to be developed manually, increasing its complexity. However, further support for NS2 had been stopped, and it was advisable to use NS3 instead in order to ensure the relevance and usability of the experiments.
  - Steep Learning Curve: use of the NS3 tool required preliminary knowledge of certain radio technology theory, such as the workings of TCP/UDP sockets and their lifecycle, and the technical differences between 2.4 GHz and 5 GHz WIFI bands with respect to transmission distance, etc. This requirement caused a few delays in the simulation phase of the study.

6.3 Source Code Framework

Based on the simulation environment, i.e. NS3, the source code development language was C++, and thus, was inherently based on the object-oriented programming methodology (OOP). The high-level class diagram was presented in figure 4.9. Topologies are developed as classes and inherit from the top Topology parent class. Thus, common characteristics of topologies, such as *fieldSize*, *samplingRate*, *topologyType* etc, and behaviours, such as *buildTopology* and *startTransmitting*, were built into the parent *Topology* class. The sub-classes, such as *ClusterTopologyApplication* and *TreeTopologyApplication,* were then developed to hold specific topology and technique behaviour. For instance, the techniques of Leach and Heed, both cluster-based topologies, were implemented in the ClusterTopologyApplication class using methods, which defined their different behaviours. Thus, when the Leach technique was simulated, its constraints were applied and its specific methods called, such as including its limitations of the percentage of cluster heads in the network, as well as its random heads selection approach, which are quite different from the approach of the Heed technique, whose cluster heads select considers the remaining energy of the sensor nodes. In this way, new techniques could be implemented by simply adding in their specific algorithms as methods, and setting their unique property values, which have already been inherited from the parent Topology class. The source code thus enables the following capabilities, both for this study and future study by other researchers.

1. It provides a framework that enables the inclusion of an unlimited number of WSN data aggregation techniques easily by simply setting values for the parameters and creating a new method to define the behaviour of the technique. This provides a possible solution to a problem that was realised during literature study, where it was discovered that there was a lack of source code for ordinary techniques, such as Leach, Heed and Pegasis.
2. The source code provides a ready-made environment for further study within the context of this research subject. This becomes useful for the sake of enhancing the model developed in this study to enhance its results by generating more data or to extend its capabilities by including more techniques.
3. In addition to 2, due to the capabilities of development environment, it is also possible to integrate the intelligent model (or other model) directly to the simulated WSN environment, in order to directly apply the recommendation of the intelligent model into the network, while also providing a real-time visualisation component to show the impact of the recommendation.

4. It uses a popular wireless sensor network research environment, that is NS3, for implementation of the framework, which especially stands out of the available simulators because it uniquely enables parameterised experiments, which run within a fast development environment, that is C++.

6.4 Intelligent Model

This section covers the intelligent model's implementation based the tools mentioned in the last section. It covers the implementation, data pre-processing stages, training, testing and evaluation.

6.4.1 Implementation

This section discusses the implementation details of the intelligent model. The ML model consists of a multi-layer artificial neural network (ANN), which initially had three hidden layers, including its input and output layers and applies *ReLU* and *SoftMax* activation functions on appropriate layers. The initial configuration of the ML model was guided by values used in corresponding experiments (Almiani et al., 2020; Diro and Chilamkurti, 2018, 2018; Hasan et al., 2019; Otoum et al., 2019).

Table 6.3 provides some detail on the initial settings for various hyper-parameters of the model.

Table 6-3 – Initial configuration for the hyper-parameters for the machine learning model. the eventual settings are discussed in the next chapter.

No	Parameter	Description
1.	MLAlgorithm	*Multi-layer Artificial Neural Network (ANN)*
2.	Programming Language	*Python 3.6*
3.	Modules and Frameworks	*TensorFlow 2.0 / Keras*
4.	Input layer units	*10 (this was modified based on the included attributes, thus serving as features)*
5.	Output Layer Units	*4*
6.	Number of hidden layers	*3*
7.	Hidden layer neuron count	*Layer 1 – 340, Layer 2 – 512, Layer 3 - 240*
8.	Activation Functions	*ReLU (Hidden Layers)* *SoftMax (Output)*
9.	Optimizer	*Adam, RMSProp*
10.	Number of samples	*216,000 samples*
11.	Epochs	*1,000*
12.	Batch Size	*10,000*
13.	Learning Rate	*0.03*

6.5 Accuracy Tests

A few accuracy tests and benchmark metrics were selected for evaluating the intelligent model. The separate instruments are discussed the following section.

6.5.1 Accuracy Score

The accuracy score methods are described below. A transformation of this is used to evaluate every step of the training process. These instruments are included by standard in the Keras framework (Bangotra et al., 2018).

- Root Mean Square Error (RMSE) - represents the standard deviation of the prediction errors. This indicates the spread of correct predictions with respect to the right predictions. It is represented by the following equation.

$$RMSE = \frac{1}{n}\sqrt{\sum_{j=1}^{n}\left(y_i - \hat{y}_j\right)^2} \qquad \textit{Equation 42}$$

  where n is the sample size, j is used as a counter, y_i represents the right prediction, y_j represents the prediction, and $\left(y_i - \hat{y}_j\right)^2$ represents the squared difference between the right prediction and the predicted value.

- Mean Absolute Error (MAE) represents the error between the predicted and the right prediction. It is represented by the following equation.

$$MAE = \frac{1}{n}\sqrt{\sum_{j=1}^{n}\left|y_i - \hat{y}_j\right|} \qquad \textit{Equation 43}$$

  where n is the sample size, j is used as a counter, y_i represents the right prediction, y_j the predicted value, and $\left|y_i - \hat{y}_j\right|$ represents the absolute value of the difference between the right and predicted values.

The accuracy plots of the machine learning model, based on a set of hyper-parameter settings, are discussed in more detail in the next chapter, where the values generated are plotted in a graph for easier assessment.

6.6 Evaluation Metrics

The evaluation of the intelligent model is performed in the next chapter. It is performed by submitting various selected application scenarios, while the recommendations are evaluated based on the ground truth. In the next chapter, various metrics shall be used in the initial evaluation of the performance of the intelligent model. This section discusses the details of some of these metrics. The *confusion matrix* as well as the *Precision and Recall,* are two common benchmarks used in the evaluation of machine learning models (Bangotra et al., 2018; Hooda et al., 2018). The confusion matrix metric classifies model responses into TP (true positive, FP (false positive), TN (true negative, and FN (false negative, a combination that dictates the performance of the model (Otoum et al., 2019). The equations used for defining these benchmarks are obtained by the following computations:

$$Accuracy = \frac{TruePositive + FalseNegative}{Total\ Number\ of\ Samles} \qquad \text{Equation 7.1}$$

The Precision and Recall metrics are also computed using equations 7.2 and 7.3.

$$Precision = \frac{No\ of\ Correct\ Positive\ Results}{No\ of\ Classifier\ Positive\ Results} \qquad \text{Equation 7.2}$$

$$Recall = \frac{No\ of\ Correct\ Classifier\ Positive\ Results}{No\ of\ All\ Relevant\ Samples} \qquad \text{Equation 7.3}$$

The next section discusses the software prototype application.

6.7 Software Prototype

This section covers the design of the software prototype and discusses its architecture and implementation, including its UI and process flow. As discussed earlier, the evaluation of the intelligent model already captures the functionality that would have been performed via the prototype, since it also intrinsically covers for singular scenarios.

6.7.1 Software Implementation

A high-level illustration of the software prototype workflow is shown in figure 6.1. It shows the important components and the flow of communications between them. The user interface provides specific fields with drop downs based on a set of expected values for a specific scenario.

Figure 6.1 illustrates the flow of the application. The circled numbers represent the different stages of the process and enable a step-by-step interpretation of the process flow. This figure contains other components that the current user interface does not include, such as an integration with a live or virtual WSN network. Circle 1 (stage 1) represents the reporting or user monitoring station, where queries or requests are submitted to the ML model. Stage 2 represents the set of requirements that have been submitted by the user, or that have been collected from an application's context. Stage 3 represents the intelligent model. The details of the appropriate input parameters are discussed in table 6.4. The output of stage 3 is the recommended technique.

The Recommended Technique box holds the best technique for consumption by the network or the user interface. It could be represented by publish/subscribe system, or as a distributed service across the nodes in the network. However, the diagram illustrates that a user can enter the details of a selected WSN network application, and the model can dynamically predict the best technique for the scenario. The diagram also indicates that the user should be able to visualize the impact on the network based on the entered details for the scenario and based on the selected network objective (i.e., energy, bandwidth, or latency in this case). In the figure, box 5 represents the network of nodes. These are expected to receive an update for the best technique given the conditions and being smart devices, are expected to self-configure themselves to use the recommended technique.

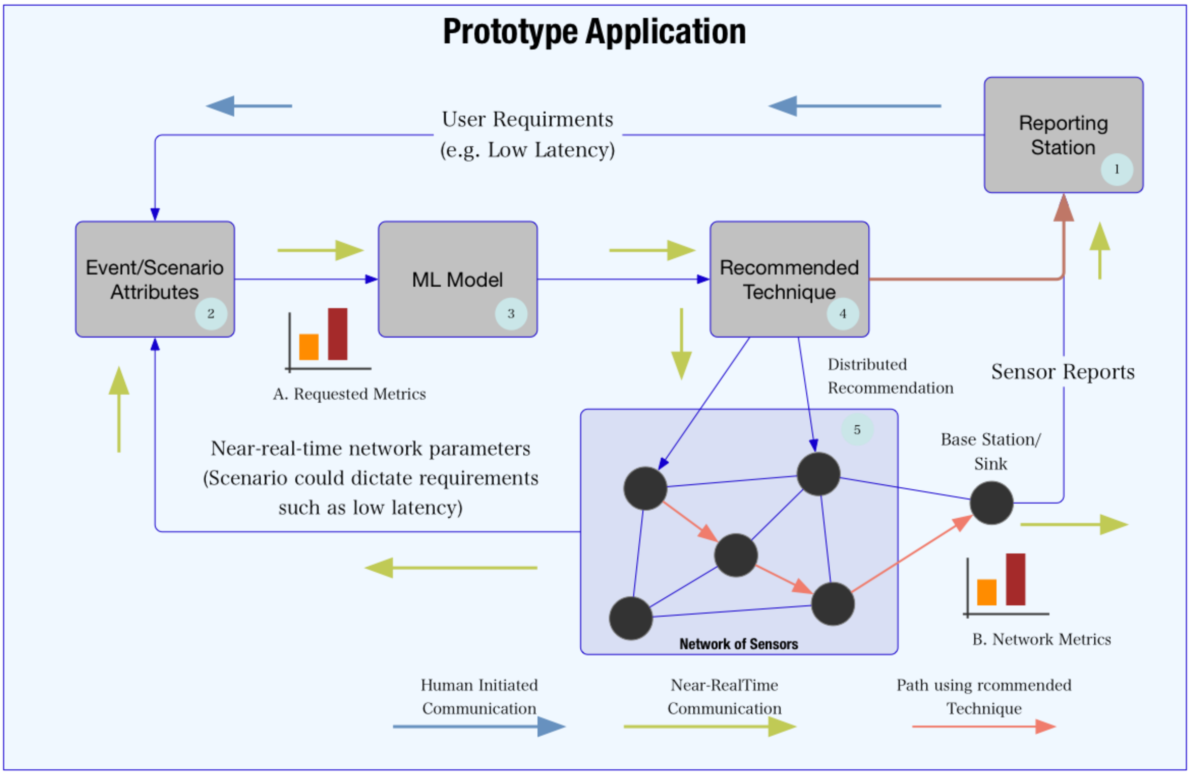

Figure 6-1 - Architecture of the prototype software application. This includes the intelligent model being used in a typical scenario query situation to obtain a recommendation for the best data aggregation technique.

6.7.2 Application User Interface

The user interface of the software consists of a web interface with certain fields, where a user is expected to select options to submit a query to the model. The user is also able to visualize the recommendations from the model. The expected fields on the interface are discussed in table 6.4. Rows 10 to 30 are alternative fields that could also be included in future improvement to the model where more features could be added to improve the training of the model. These are not included in the training feature list.

Table 6-4 - Description of fields shown on the ML model query UI (and used in intelligent model training). The table includes all fields identified as applicable for training the model, while the selected fields used to train the model are shown in bold. The selected fields were found to have high correlation to the technique, as well as being measurable easily for entering into the user interface.

S/N	Field Name	Description	Included for Prototype UI
1.	Scenario Name	Enables a label for the submitted scenario or sample.	Included
2.	Number of nodes	This represents the average number of nodes in a scenario. It is expected that the number of nodes should be equivalent to the size of the network, unless the nodes have high-powered sensors or otherwise, the network consists of heterogenous nodes with various capabilities	Included
3.	Sampling Rate / Interval	The average sampling rate or interval represents the expected the sampling rate expected in a selected scenario. This could change based on the event state or classification as defined in 2.	Included
4.	Objectives (Energy Consumption, Bandwidth Consumption, Latency)	Accepts a value that indicates the importance of each attribute to the application context.	Included
5.	Packets	This represents the number of packets sent in communication between the sensor nodes	Included
6.	Packet Size	This represents the size of the packets send between nodes in the network	Included
7.	Initial Energy	This represents the initial energy of the nodes in the network	Included
8.	Field Size	This represents the size of the deployment field of the network of nodes	Included
	The next set of variables were also identified as plausible entry values for the model, however for future enhancements to the model. They represent valid data that could be used to train as well as query an intelligent model serving the same purpose as that developed in this study. This study does not, however, include them.		
9.	Classification	Indicates one of two states in which the scenario can be categorised – Detection or Monitoring. An event moves from monitoring to detection when a trigger threshold has been reached.	Not Included

10.	Physical Sensor Topology	Selection of the physical distribution of the sensors (i.e. Random Sparse, Random Dense, and Ordered Linear)	Not Included
11.	Physical Network Topology	Selection of the closest physical network topology to the scenario (i.e. Star, Mesh, Bus, etc.)	Not Included
12.	Physical Network Size	Represents the average size of the network. Values used include Room, Apartment Block, City, in order to provide an average size representation to the user	Not Included
13.	Sink/Base Station Distance	Represents the distance of the reporting station or base. Like physical network size, the values used include Room, Apartment Block, City, in order to provide an average size representation to the user. This feature could also be referred to as sink reception distance.	Not Included
14.	Environment	This determines the immediate environment of the network. For instance, it attempts to distinguish deep sea monitoring from ground-surface or in-space monitoring, which all have different characteristics.	Not Included
15.	Variable	This attempts to identify commonalities between scenarios, such as gas for CO_2, NO_2, CO monitoring, temperature for body temp, air temp monitoring, etc.	Not Included
16.	Location Awareness	This toggle indicates whether the application or scenario requires location awareness, thus, requiring location awareness among nodes	Not Included
17.	Aggregation Type	Selection of one of the prominent aggregation functions (i.e. ADD, DIV, COUNT, MAX, etc.). This is usually defined by the scenario or application use case.	Not Included
18.	Minimum Energy consumption	Defines the relevance of energy consumption as an objective to the scenario, thereby highlighting this to the ML model	Not Included
19.	Minimum Bandwidth consumption	Defines the relevance of bandwidth consumption as an objective to the scenario, thereby highlighting this to the ML model	Not Included
20.	Minimum Latency	Defines the relevance of latency as an objective to the scenario, thereby highlighting this to the ML model	Not Included
21.	Required Connectivity	Represents a binary list of connectivity requirements (i.e. Partial, and Full). Some scenarios require partial connectivity among nodes (e.g. earthquake) especially when they are homogenous nodes, while others require full connectivity (health monitoring) especially when they are heterogenous nodes.	Not Included
22.	Communication Algorithm	Represents the structure of communication, i.e. one of Hierarchical, Flooding or Diffusion. This is a characteristic of the DA technique used in a scenario.	Not Included

23.	Sensing Trigger	Represents the type of trigger relied on to trigger data collection in the scenario. It could have values o Query, Event, Real-time or Hybrid. This is characteristic of the scenario.	Not Included
24.	Homogenous Nodes	The binary toggle indicates if the set of notes are of whether the network consists of homogenous nodes. This is default for this study	Not Included
25.	Periodic Reporting	Indication of whether periodic reporting is used	Not Included
26.	Event Reporting	Indication of whether event reporting is used	Not Included
27.	Location Awareness	Indication of whether location awareness is required	Not Included
28.	Node Mobility	Indication of whether node mobility is required	Not Included
29.	Sink Reporting Mode	Selection of the physical mode of reporting to the sink (for instance, Many-to-one in earthquakes via sensor to satellite links, and one-to-one in undersea monitoring systems via linked sensor networks)	Not Included

Figure 6.2 presents a snapshot of what the web-based client would look like. It features only the selected fields, which will need to be filled in by the user.

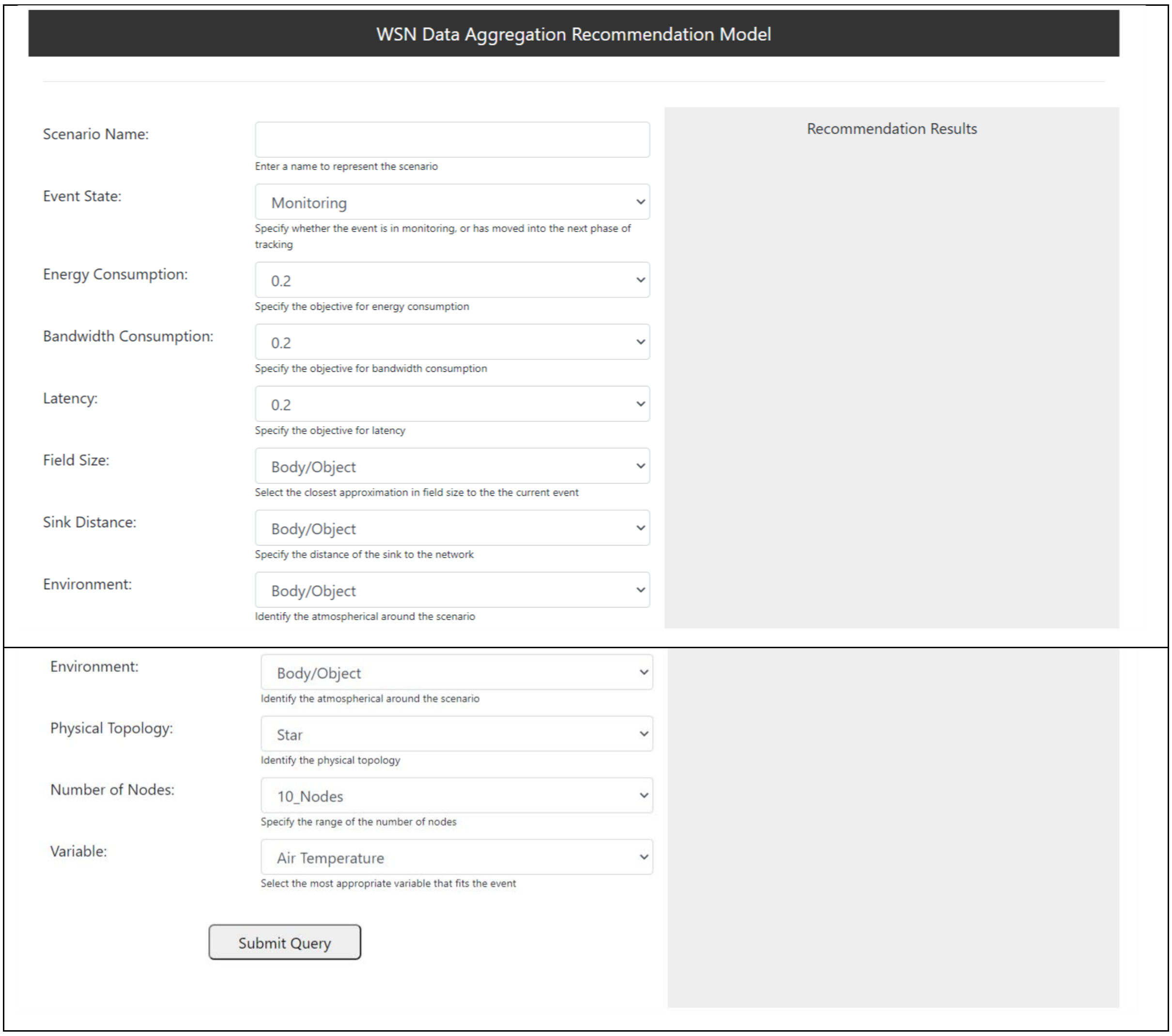

Figure 6.26-2 - Web-based user interface for the prototype application, which consumes services from the intelligent model.

6.7.3 Application User Process Flow

This section discusses the proposed workflow of the software prototype from the user's perspective, and according to figure 6.2. This is split into two flowchart diagrams in figure 6.3 to enhance visualisation. The flows include the query submission and subsequent recommendation.

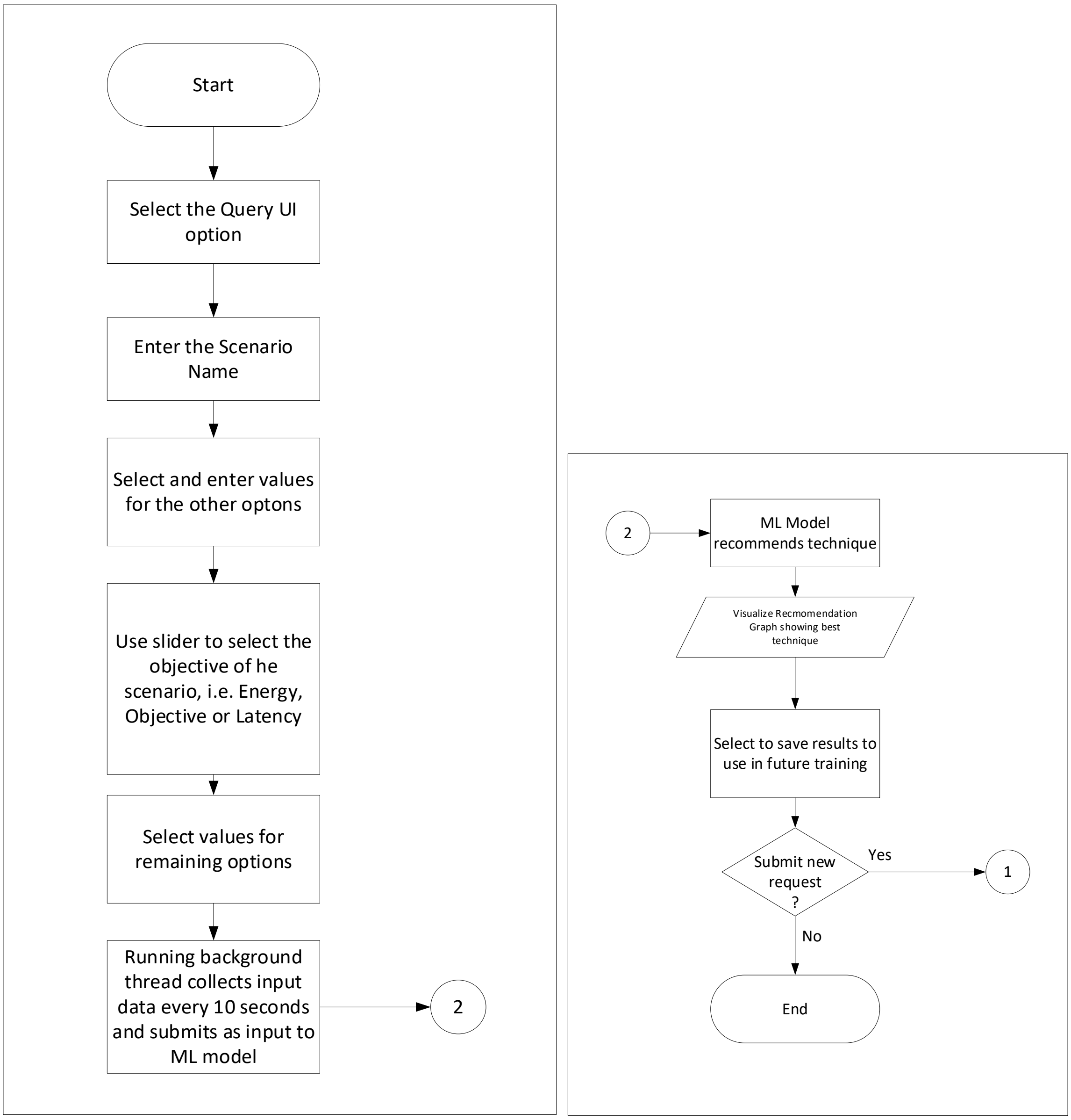

Figure 6.3 – Flowchart diagrams 1 and 2 showing the user experience process flow for the application. In the left flowchart, the user selects options provided on the interface. A background thread runs to collect the current entries and submits to the ML model for recommendation. The results are presented in the recommendation results interface.

The above flows are prescriptive process flows for the prototype software that runs above the intelligent model, enabling a human user to query the model for a technique recommendation. This could be adjusted for real-time query from a realistic wireless sensor application where the services are provided via REST APIs.

6.8 Summary

This chapter has covered the implementation design of the intelligent model and software prototype. It presented the details of the hardware and software used in the implementation, highlighting the constraints and challenges faced with the tools. The contribution of the source code, which forms a framework for the exploration based on this study, was also highlighted. Then it discussed the implementation of the machine learning model, discussing in detail how the training data was computed. Then it presented the training hyper-parameters and the actions taken to obtain optimum values for the training. Then architecture of the software prototype, its user interface, and process flows are also presented. The next chapter covers the build, testing and evaluation of the intelligent model. In summary, this chapter accomplishes the contribution of *Prototype Framework and Source Code.*

7 Intelligent Model Testing and Evaluation

7.1 Overview

This chapter discusses the evaluation of the intelligent model based on various single and complex scenarios. The design and development of the intelligent model was discussed in chapters 4, 5, and 6. The last chapter covered the implementation design of the intelligent algorithm, the hardware and software environment and the software prototype. This chapter discusses the evaluation of the intelligent model based on several WSN application scenarios and presents these results in corresponding graphical plots.

7.2 Scenario Use Cases and Features

The training of the intelligent ML model was based on a set of features. For completeness, these features are also listed below:

- Objective: the objective of the WSN application
- Number of Nodes: the number of participating nodes in the application
- Field size: the field size of the application
- Packet size: the size of packets used in communication
- Packets: the number of packets transmitted between nodes
- Time: the time instance of the data sample
- Technique: he best technique applicable to the instance
- Benefit Value: the benefits value for selecting the best technique over the rest

During the lifetime of wireless sensor network application, a few features could remain constant, such as the number of packets and the packet size. However, due to various environmental factors, such as a change in sapling rate requirements, any of these could change mid-lifetime. The intelligent model is expected to detect these changes and to make recommendations dynamically. The benefit feature in the data, as shown in the list above, represents the benefit of selecting the best technique, if compared to the selected technique in the data. This implies that while a non-optimal technique is in operation, this field represents the value or benefit of switching onto the best technique for that scenario.

7.3 Intelligent Model Testing and Evaluation

This section discusses the training process of the model and provides certain benchmark scores. As a reminder, it was mentioned in the last chapter that the Python *Keras* framework was used to build the intelligent model using an Artificial Neural Network (ANN).

7.3.1 Dataset Details

Table 7.1 describes the statistics of the data used in training, testing and evaluating the intelligent model. This refers to the unique data samples, and the number of scenarios that were represented in the data. The distribution of the dataset is described in table 7.2.

Table 7-1 - Intelligent Model Training Data Information. The data used for the various test and evaluation use cases represent unrepeated data from the data that was withheld from the original data sample, and that was not used in the training process.

Data Summary / Use Case	Value
Total Data Samples	*43,179 samples*
Total Unique Scenarios	*905 scenarios*
Data Per Scenario	*Approximately 50 records per scenario*
Model Development	*41,117 (of total 43,179 samples)*
Model Training Samples	*32,890 samples (of 41,117 samples)*
Model Testing Samples	*8,227 samples (of 41,117 samples)*
Evaluation Samples	*2,062 samples (of total 43,179 samples)*

Table 7-2 - Set of experiments carried out to evaluate the performance of the intelligent model

No	Experiments	Description
1	Single Scenario/Metric Evaluation (Energy Consumption)	*50 samples – a single scenario that was evaluated based on a WSN application objective of energy consumption*
2	Single Scenario/Metric Evaluation (Bandwidth Consumption)	*50 samples – a single scenario that was evaluated based on a WSN application objective of bandwidth consumption*
3	Single Scenario/Metric Evaluation (Latency Consumption)	*50 samples – a single scenario that was evaluated based on a WSN application objective of latency*
4	Multiple Scenario Evaluation (3 scenarios)	*150 samples (50 x 3) – a scenario of 3 combined scenarios used to evaluate the model*
5	Model Performance Evaluation	*980 samples – large sample used in evaluation to compare the model performance to the state of art and the use of a single technique throughout the WSN lifetime.*

6	Realistic Scenario Evaluation	*3 unique scenarios representing realistic environments used to evaluate the model as done in No. 5*

The rest of the chapter covers the details of the experiments carried out to evaluate the intelligent model as discussed in table 8.3.

7.3.2 Hyper-Parameters Details and Model Topology

Table 7.3 presents the final hyper-parameter values used to train the machine learning model. These were finalised after applying a grid search process using Python *Keras' GridSearch*. The hyper-parameters used to train the model are presented in table 7.3 below:

Table 7-3 - Final hyper-parameters used for the intelligent model

No	Parameter	Description
1.	Input layer units	*11*
2.	Output layer units	*4 - (4 binary units used to represent the four techniques – Leach, Heed, Pegasis and Dbst)*
3.	Number of hidden layers	*3*
4.	Hidden layer nodes specification	*Layer 1 – 500, Layer 2 – 1000, Layer 3 - 400*
5.	Activation Functions	*ReLU (Hidden layers)* *SoftMax (Output layer)*
6.	Optimizer	*Adam*
7.	Number of samples	*41,117 samples*
8.	Epochs	*600*
9.	Batch Size	*1000*
10.	Learning Rate	*0.1*
11.	Initializer	*uniform*

Table 7.4 presents the details of the model after the build process.

Table 7-4 - Details of the model after build, later used in the training process

Layer (type)	Output Shape	Param #
Input layer	*(11 nodes)*	
dense_1 (Dense)	*(None, 500)*	*6000*
dense_2 (Dense)	*(None, 1000)*	*501000*

dense_3 (Dense)	*(None, 400)*	*400400*
dense_4 (Dense)	*(None, 4)*	*1604*
Total params: 909, 004 Trainable params: 909, 004 Non-trainable params: 0		

7.3.3 Model Training, Accuracy and Benchmark Measurements

Once training was performed based on the above specifications, an accuracy trend graph was generated. This is presented in figure 7.1. the graph represents a starting gradual learning phase, which became steeper in the middle stage of the training process. Beyond the middle stage, the training process started to level off to a plateau albeit a few perturbations. Noticeable are frequent spikes in the training process that indicate a sudden drop in accuracy. It was suspected that the reason for this was based on the use of a feature that was uncorrelated with the rest of the data, or perhaps an outlier. However, the subsequent performance of the model was considered enough for the purposes of this study.

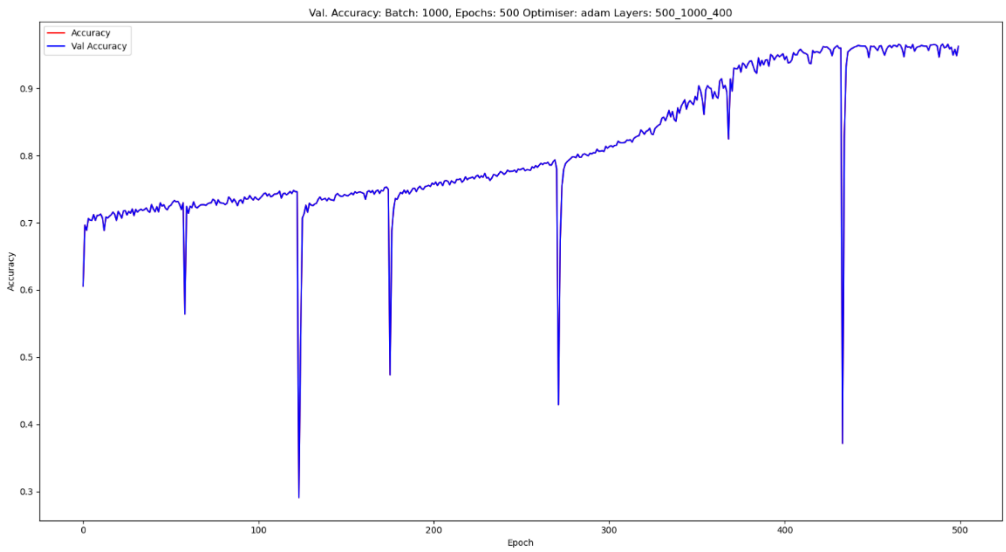

Figure 7-1 - Training accuracy progression plot for the intelligent model. Training starts slowly and gradually climbs into a plateau with a few sharp perturbations.

The following tables present various benchmark metrics of the intelligent model. Table 7.5 presents the confusion matrix of the model and indicates that the model does have high scores for predictions for each class, compared to wrong predictions.

Table 7-5 - Model evaluation by the Confusion Matrix with 980 samples (using Python Scikit-learn library)

Benchmark	Dbst	Heed	Leach	Pegasis
Dbst	*221*	*7*	*11*	*5*
Heed	*7*	*224*	*5*	*8*
Leach	*11*	*8*	*219*	*9*
Pegasis	*5*	*12*	*7*	*221*

Table 7-6 - Computations based on Confusion Matrix

Benchmark / Evaluation	Computation Algorithm	Result
Accuracy	*(TP + TN) / Total* *(221 + 224 + 219 + 221) / 979*	*0.903*
Error Rate	*(FP + FN) / Total* *95 / 979*	*0.097*
Classification Report	*Not applicable*	*0.870750273145905*

Table 7.7 shows the Cohen's Kappa score, which depicts a measure of how well the model performed compared to how well it would have performed simply by chance. These scores were generated from the Python Scikit-learn library. The scores shown in table 7.7 indicate that the model does perform well in terms of accuracy, which corresponds to the accuracy score in table 7.6. The scores for the precision shows that the model has a high score for correctness when the recommended technique happens to be the ground truth. The recall scores also indicate a high accuracy for the model when it recommends the exact class when this happens to be the ground truth.

Table 7-7 - Model evaluation based on Cohen's Kappa Score (Python Scikit-learn library)

Class	Precision	Recall	F1-Score	Support
Leach	*0.90*	*0.89*	*0.90*	*247*
Heed	*0.89*	*0.92*	*0.91*	*244*
Pegasis	*0.91*	*0.90*	*0.91*	*245*
Dbst	*0.91*	*0.91*	*0.91*	*244*
Average	*0.90*	*0.90*	*0.90*	*980*
Macro Avg.	*0.90*	*0.90*	*0.90*	*980*

Weighted Avg.	*0.90*	*0.90*	*0.90*	*980*

The rest of the chapter presents plots illustrating the performance of the model.

7.4 Model Evaluation

This section discusses the evaluation of the model with respect to selected scenarios. The plots of the evaluations using the selected scenarios are shown later in this chapter. Most of the test data used in the following experiments are non-repeating. The experiments were carried out by submitting scenario data to the model in a "input-stream" form, while the model provided recommendations in an "output-stream" form, which could then be scored based on accuracy.

7.4.1 Baseline Performance

The following evaluations shall rely on a base performance defined by state-of-the-art. As a reminder, the state of the art consists of the current data aggregation approach used in WSNs. which consists of the use of a single DAT through the lifespan of an application, irrespective of changing context. Thus, the base performance is that obtained in such scenarios where the "preferred" technique is used across the WSN for a given scenario. This shall be used as reference in the following evaluations and usually be presented in comparison to the performance obtained from the application of the intelligent model.

7.4.2 Energy Consumption, Bandwidth Consumption and Latency

This section covers the evaluation of the model based on chosen network metrics, which include energy consumption, bandwidth, and latency. The dataset used in this experiment consisted of 3 randomly selected scenarios from the dataset. The accuracy scores represent the score obtained based on data from the scenarios. The experiment is used to represent random ubiquitous application scenarios where the objective changes over time during the lifetime of the application. The plots are based on the timescale of the event. The flat tops of the graphs indicate accurate prediction, while drops indicate inaccurate predictions. Thus, the graphs were plotted based on accurate vs inaccurate predictions, where accurate predictions were given a value of 1, while inaccurate predictions were given a value of 0.

Figure 7.2 shows the model's performance based on the metric of energy consumption. The figure indicates that an accuracy of 79.310% was obtained for this evaluation. Following this, figure 7.3 shows similar evaluation for bandwidth consumption with an accuracy score of 83.33%, while figure 7.4 shows the same evaluation for latency with a score of 77.083%.

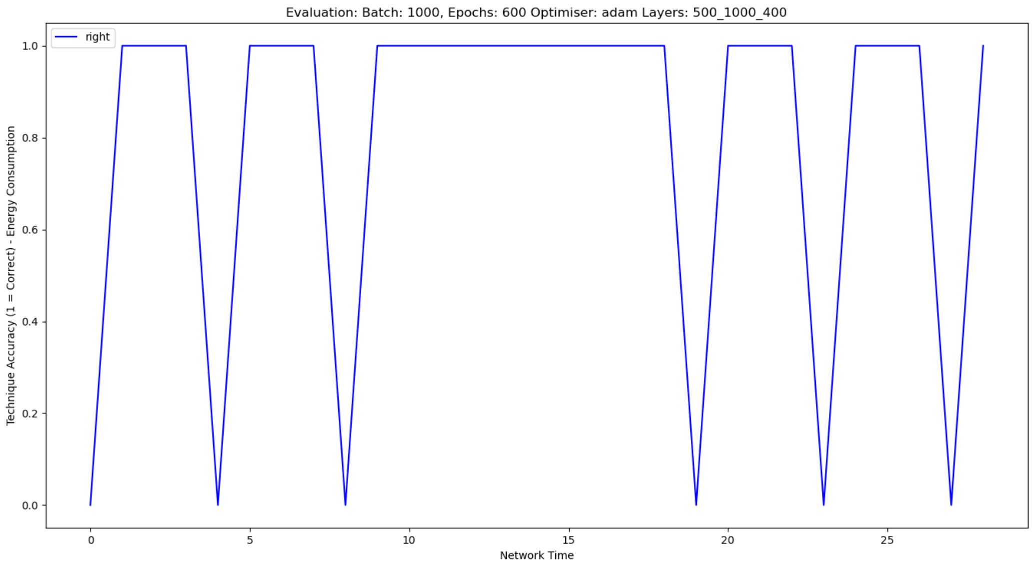

Figure 7-2 - Plot of model performance for single scenario for energy consumption. Accuracy: 79.310%

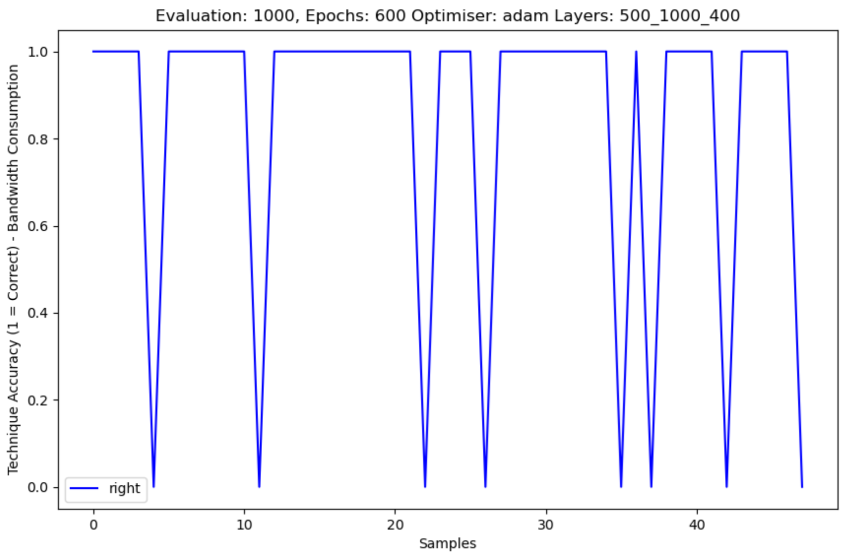

Figure 7-3 – Plot of model performance for single scenario for bandwidth consumption. Accuracy: 83.33%

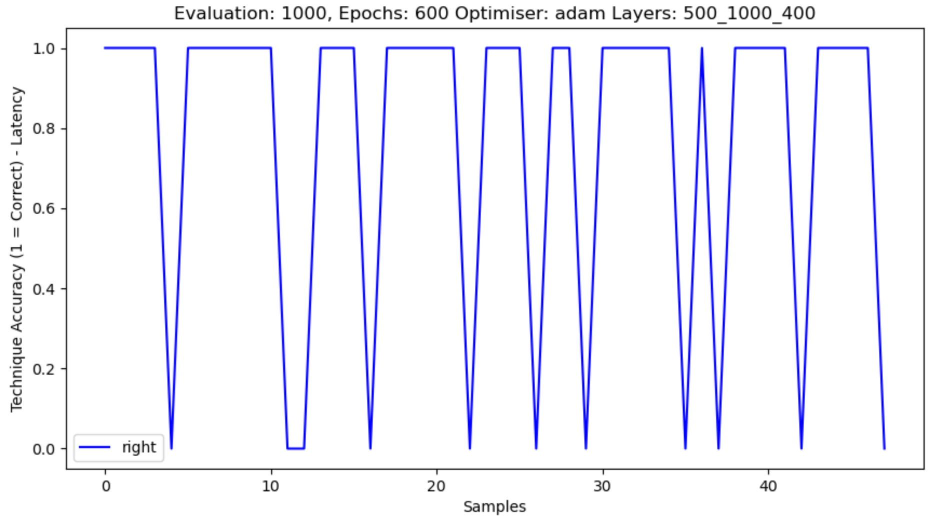

Figure 7-4 – Plot of model performance for single scenario for latency. Accuracy: 77.083%

Figure 7.5 shows the performance of the model based a combination of three different scenarios, consisting of a change in the number of nodes and a change in the interval or sampling rate. Such scenario data could be used to represent a change in the number of nodes while the event is active, and then a change in the sampling rate, such occurs when a lower frequency event (an Earthquake for instance) triggers need for a higher frequency event (a Tsunami). As is observable from the accuracy score, the model performs substantially better than the base performance.

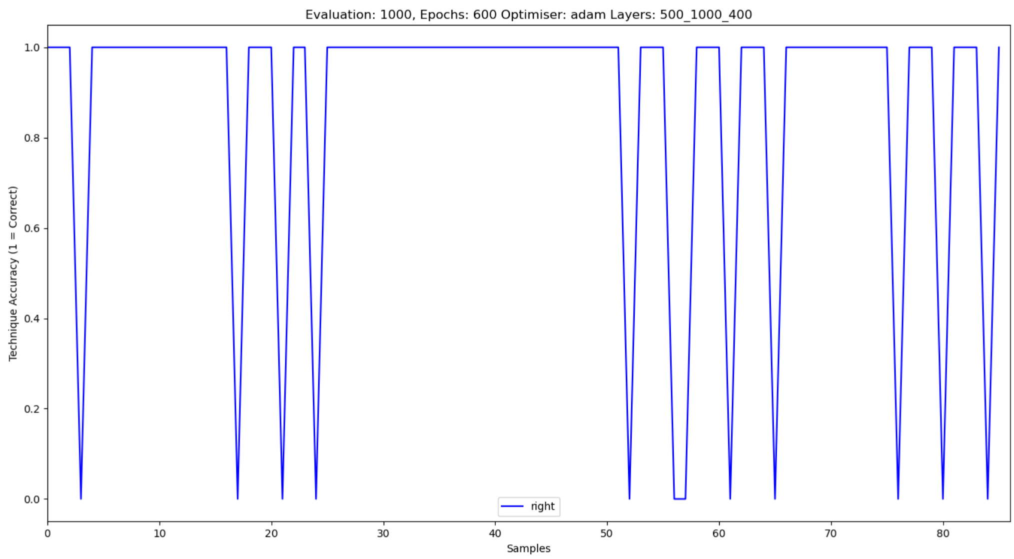

Figure 7-5 - Plot of model performance for multiple scenarios with changing data: nodes (10, 20), intervals (0.01, 0.001, 0.0001), packets=20, packet size=64, initial energy=1J, field size=50 Accuracy: 86.046%

Figure 7.6 presents the performance of the model based on a different set of random selection of the dataset consisting of four different scenarios specifically selected based on varying scenario objectives. The model achieves an accuracy of 90.306% in detecting the best technique based on a combination of all the objectives. Only a subset of the data is shown in this figure to reduce the clutter and to present a visual representation of the performance.

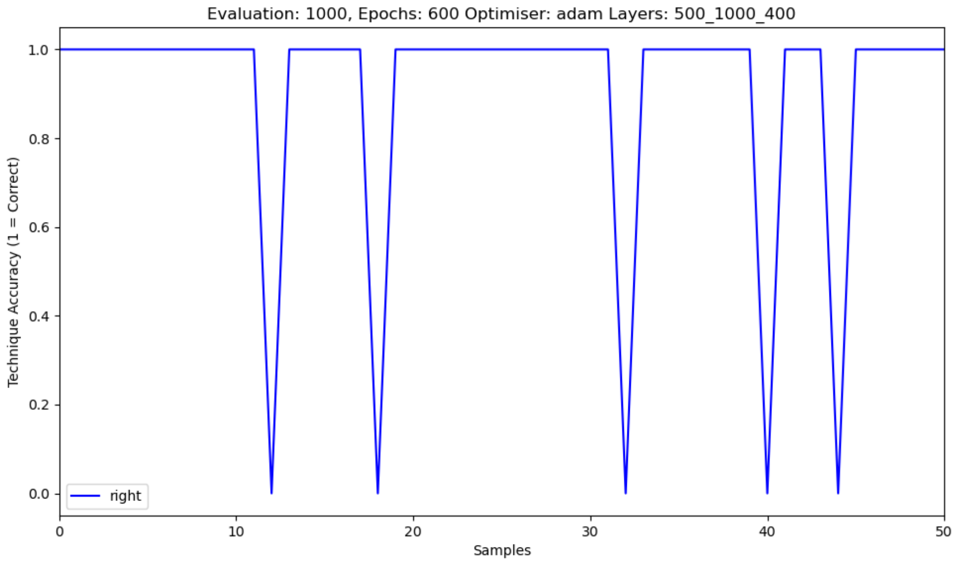

Figure 7-6 - Model performance for selected evaluation data. X-axis indicates scenario time instances. Accuracy: 84.5%

Figure 7.7 summarises the performance of the model based on the five discussed use cases, i.e., energy consumption, bandwidth consumption, latency, changing characteristics and set of random scenarios. None of these use cases used repeated data selected from the testing data pool. Each bar in figure 7.7 includes the accuracy of the model given the use case, as shown in the past graphs. It can be observed that the model performs better with bandwidth consumption than it does for both energy consumption and latency. However, with a score of 74.509% for energy consumption, it is assumed that its use would immensely minimise the waste in resources in using a technique that does not perform optimally in mixed scenarios.

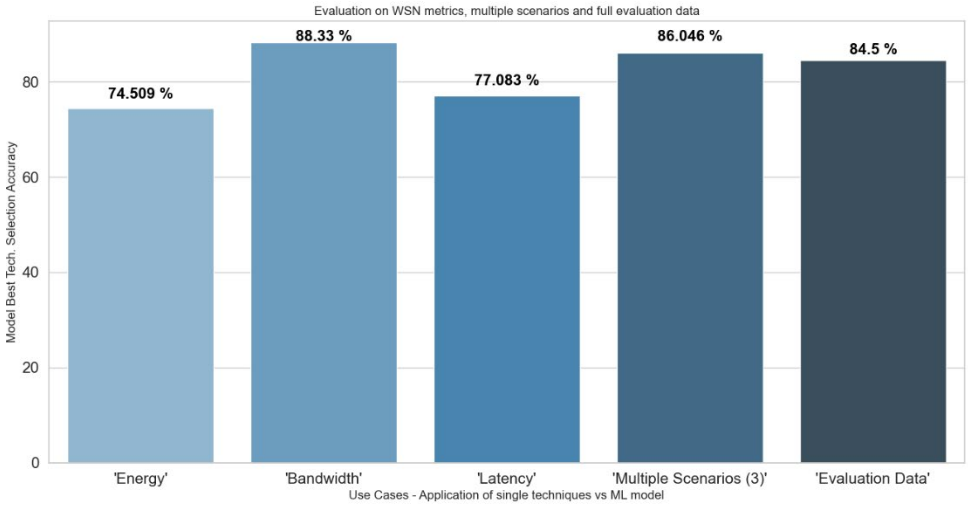

Figure 7-7 - Model performance across various use cases:
(i) energy consumption as objective, (ii) bandwidth as objective, (iii) latency as objective,
(iv) combined dynamic scenarios, and (v) separate larger random evaluation data.

7.4.3 Model Performance Compared to State-of-the-Art Approaches

The following graphs illustrate the performance of the model as compared to the current state of the art. In the state-of-the-art approach, a single technique is applied throughout the lifespan of an application. To recap discussions presented earlier in the study, this constraint is usually due to the environment in which WSNs are deployed, which could be inaccessible or uncontrollable by humans. However, during the lifespan of an application, various characteristics could change, such as the number of active nodes and the sampling rate. A single active technique used throughout the lifespan could, in different states, be found to be ineffective or to perform sub-optimally.

The following graphs present a comparison between the performance of the state-of-the-art approach against performance of the intelligent model (ML). In order to provide a visual representation of the performance, the plots are shown in bar charts.

The objective of the next set of graphs is to evaluate the model based on three specific use cases, as listed below:

- The performance based on an application of a single technique (each of Leach, Heed, Pegasis and Dbst) across the entire dataset
- The performance based when the best technique is selected for each of the scenarios

- The performance based on when the intelligent model is used to determine the best technique

Based on the above list, the plots in figures 7.8, 7.9, 7.10, 7.11, 7.12 and 7.13 represent six out of 111 scenarios, which were selected from 980 sample records. These are shown to provide a representation of the performances obtained from scenarios when the best techniques are used.

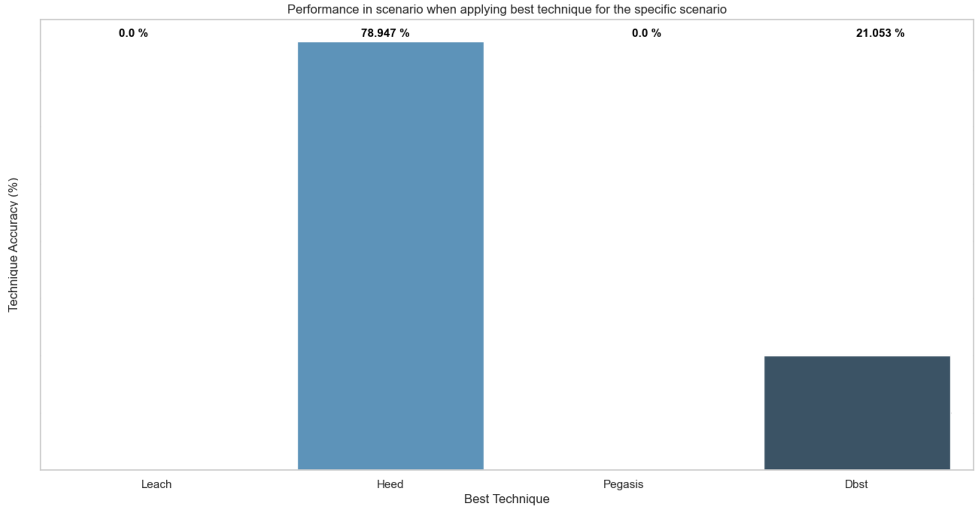

Figure 7-8 - Single scenario selected from dataset illustrating best technique performance for given scenario (nodes:10, packets:20, packet_size:128 bytes, field_size: 30.0 m^2, interval: 0.1s, initial_energy: 10J)

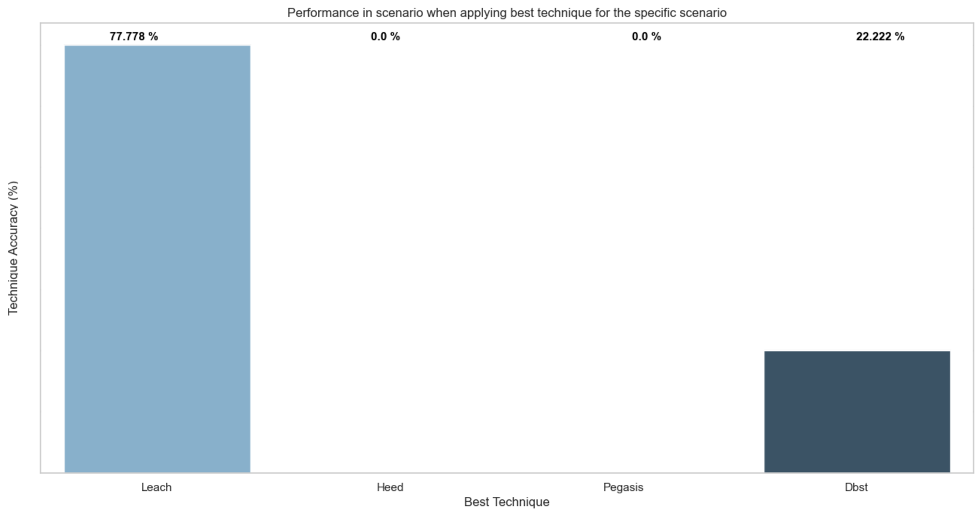

Figure 7-9 - Single scenario selected from dataset illustrating best technique performance for given scenario (nodes:100, packets:100, packet_size:256 bytes, field_size: 100.0 m^2, interval: 0.001s, initial_energy: 1J)

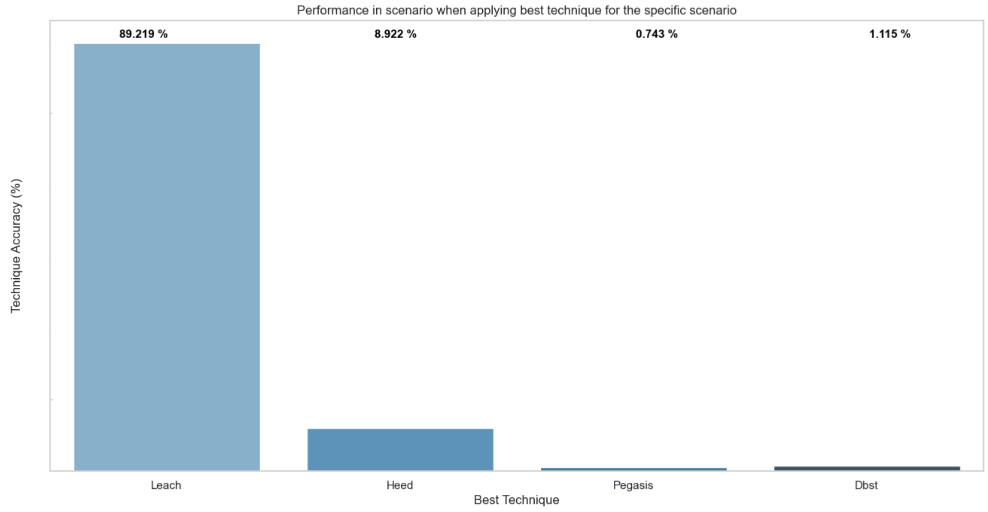

Figure 7-10 - Single scenario selected from dataset illustrating best technique performance for given scenario (nodes:30, packets:50, packet_size:128 bytes, field_size: 50.0 m^2, interval: 0.01s, initial_energy: 10J)

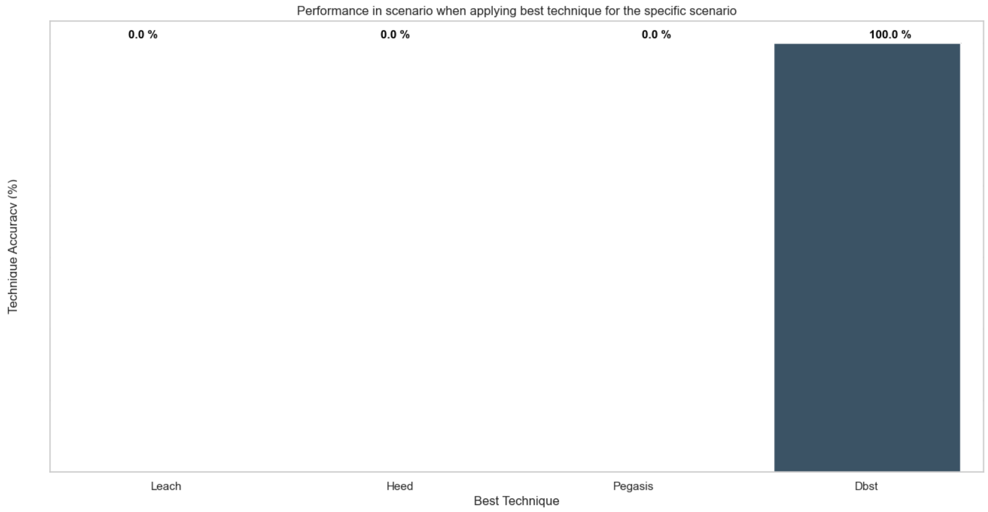

Figure 7-11 - Single scenario selected from dataset illustrating best technique performance for given scenario (nodes:100, packets:100, packet_size:128 bytes, field_size: 150.0 m^2, interval: 0.0001s, initial_energy: 10J)

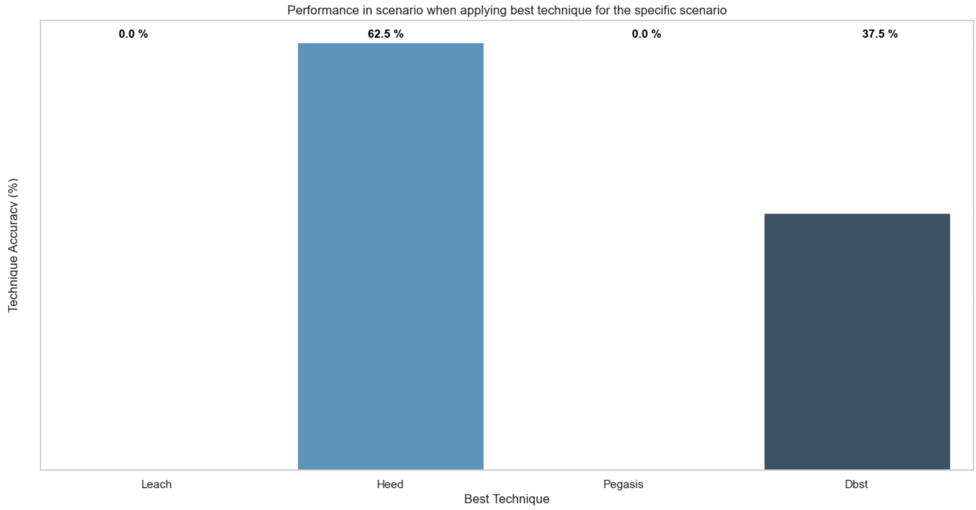

Figure 7-12 - Single scenario selected from dataset illustrating best technique performance for given scenario (nodes:100, packets:100, packet_size:512bytes, field_size: 150.0 m^2, interval: 0.001s, initial_energy: 1J)

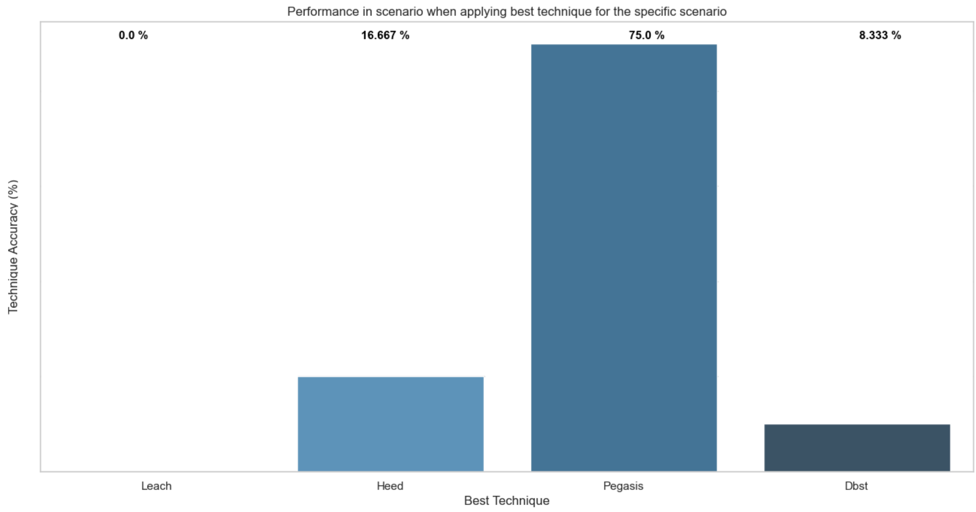

Figure 7-13 - Single scenario selected from dataset illustrating best technique performance for given scenario (nodes:50, packets:50, packet_size:1024 bytes, field_size: 100.0 m^2, interval: 0.01s, initial_energy: 10J)

It is obvious from figures 7.8, 7.9, 7.10, 7.11, 7.12 and 7.13 that the best techniques hold a higher score than the rest of the techniques in the scenarios. However, it is not in all cases that they perform at a 100% score for each scenario. This gap is one aspect that is supposed to be improved by the intelligent model.

Figure 7.14 shows the combined performance of the above graphs (based on the entire data), the application of single techniques across the entire data, and the performance obtained when the intelligent model is used. It indicates the techniques provide on average a quarter performance across the dataset, while the use of the best techniques performs as high as 70.74% combined. This represents the state of the art, where the preferred technique is selected at the start of a scenario. In retrospect, the additional limitation in the state of the art includes that the active technique cannot be changed when scenario characteristics change, a capability provided by the intelligent model.

The intelligent model provides a performance of 90.306%, which is approximately 20% higher than the performance of the state-of-the-art approach. But in addition to this performance, there is the additional benefit of dynamism and speed of selection based on using the intelligent model.

Figure 7.14 shows a combined overview of the comparative performance of all approach's above, including with the intelligent ML model.

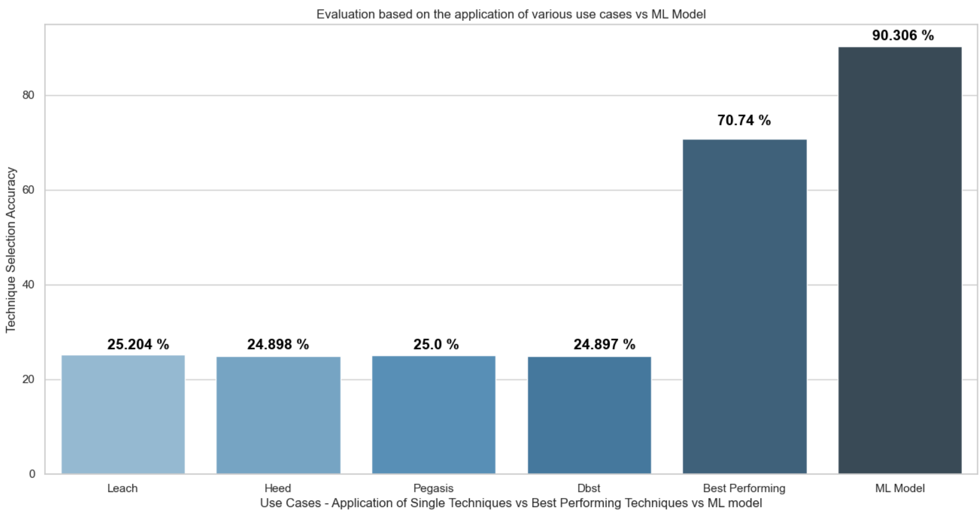

Figure 7-14 - Plot showing a comparative visualisation of the performance of various use cases – (i - iv) single technique approach (v) state of the art approach, (vi) intelligent model approach.

7.4.4 Realistic Multiple Use Case Scenario and Performance Comparison with Best Performing Technique

This section investigates the evaluation of the model based on a selected realistic use case. A similar analysis is performed as done in the last section. The term "realistic use case" as used here describes the set of values assigned to certain features, which replicate the typical situation in a realistic scenario. These consist of sampling rate (or interval), number of nodes and the field size. Other scenario parameters (i.e., initial energy, packet size, number of packets) are included to ensure that single scenarios are used in each case.

Table 7.8 presents the pre-selected values for the WSN features, which are used to model three realistic use cases. These are selected to represent three scenarios that have a relationship described by a progression of events, which change the scenario characteristics over time. The linked multiple-scenario use cases include an earthquake event, which triggers a wildfire event, and subsequently also triggers a tsunami event. It is assumed that the same nodes are deployed and exist within all scenarios, but also expanding in number as the multi-event widens in field size. it is also assumed the nodes can individually query a source for the preferred technique and select this in their configuration. Thus, table 7.8 summarises the features which represent a change in circumstances while each event transcends into the next event. Specifically, the dynamic features include the number of nodes, interval (or sampling rate), and the field size.

Table 7-8 - Realistic Use Case Analysis (Cells in light grey indicate that the values do not change)

Event	Description	Nodes	Interval	Field Size	Packets	Packet Size	Initial Energy
Earthquake	An earthquake requires a low sampling rate and slow changing field size	30	0.1 sec	100 m^2	50	128	1
WildFire	A wildfire involves quick spread of the event with increasing node involvement and increasing field size and increased sampling rate	50	0.01 sec	100 m^2	50	128	1
Tsunami	A tsunami could be triggered by an earthquake and would involve increased field size, sampling rate and more nodes.	100	0.001 sec	150 m^2	50	128	1

Figures 7.15, 7.16 and 7.17 present the performance based on selecting techniques using the state-of-the-art approach, where a dedicated technique is used in the scenario. Thus, figure 7.15 shows the performance in the earthquake event. In this figure, the Heed technique is considered the best technique and performs better than the remaining techniques with an accuracy of 71.42%. Figure 7.16 shows the same graph for the wildfire event with a score of 64.286%, while figure 7.17 shows the performance for the tsunami event with a score of 66.67%.

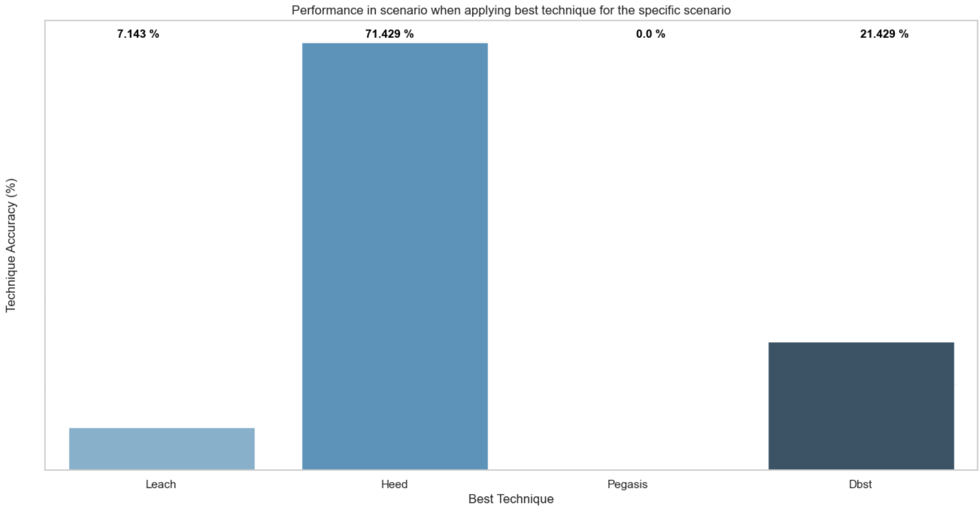

Figure 7-15 - Scenario Earthquake - Best Performing technique – Heed

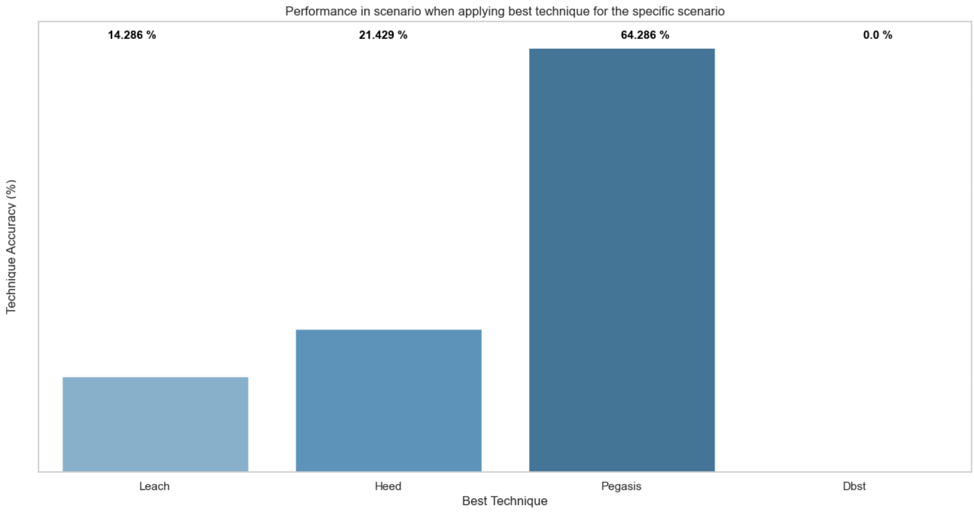

Figure 7-16 - Scenario Wildfire - Best Performing technique – Pegasis

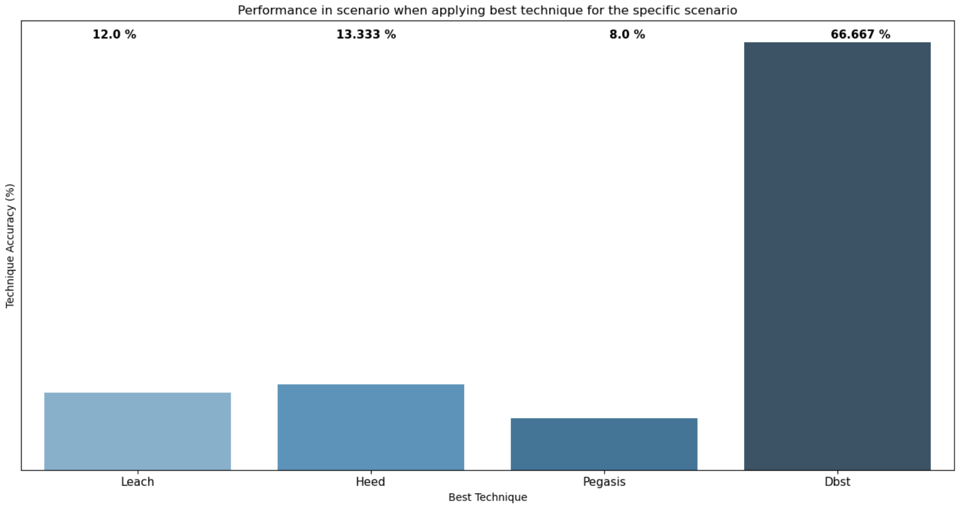

Figure 7-17 - Scenario Wildfire - Best Performing technique - Dbst

Similar to the presentation shown in figure 7.14, figure 7.18 shows the combined plots for the use cases where the performances are compared between the case where only one technique is applied across the entire 3-scenario dataset, when the best technique is applied, as shown in figures 7.15, 7.16, and 7.17, and when the intelligent model is applied across the 3-scenario dataset. It can be observed here that the intelligent model has a 24.665% improvement over the state of the art. As mentioned, this performance comes with related automated ubiquitous technique selection and handling of new scenarios.

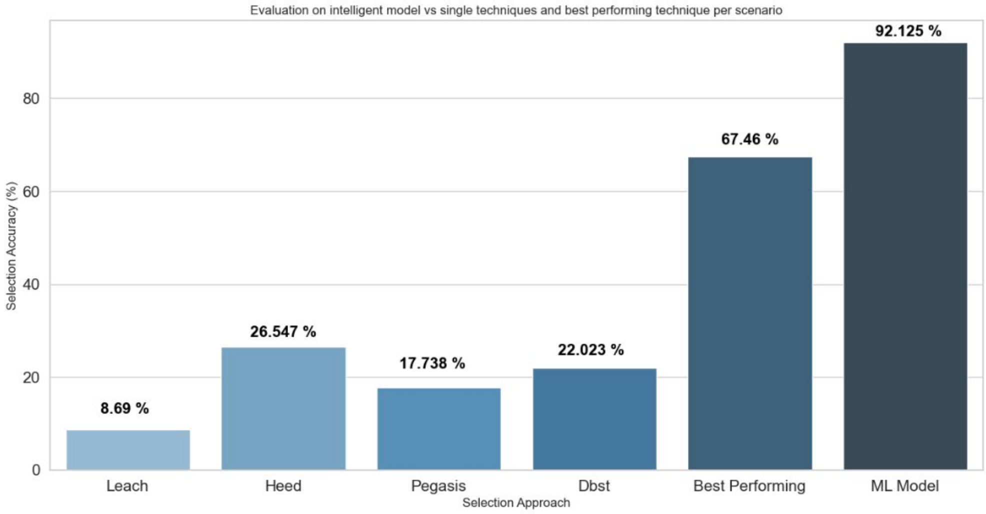

Figure 7-18 - Evaluation of performance of the intelligent model vs the use of initially selected techniques, and combined use of best performing technique across realistic use case scenarios

7.5 Summary

This chapter has covered the testing and evaluation of the intelligent model. The chapter commenced with a discussion of the statistical analysis of the dataset used in training the model, as well as the final the hyper-parameters used in training the model. Then the training accuracy plot was presented, while the model's performance based on various benchmarks was discussed. The model was evaluated based on various single and complex WSN scenarios and the results were plotted in various bar charts. These included the model's performance in recommending techniques given various network performance metrics, such as energy consumption, bandwidth consumption and latency. These were plotted against the corresponding performances based on the state of the art, where a single technique was used, and when the best technique was selected and used across the entire scenario. The model was also evaluated using data that modelled a near-realistic scenario. Substantial model performance was observed through-out the evaluations, and this implied enough justification for the effectiveness of the intelligent model.

8 Conclusions

8.1 Introduction

This chapter discusses the study's aim, and how the results obtained have been able to address the questions and objectives. Wireless sensor networks were introduced in early chapters, while their limitations with respect to data aggregation, was discussed extensively. This research explored ways to optimise the data aggregation performance in such networks by exploring their behaviour and identifying the important characteristics to be considered in addressing these problems. The research then proceeded to design and develop an intelligent model that could accomplish this objective.

Wireless sensor networks are established to sense and capture data from various phenomena in mostly inaccessible environments. The lack of the capacity to adjust based on changes within the environment forms a challenge. The aim of this research involved the development of a dynamic and adaptive intelligent model that would be able to manage a set of smart devices (such as smart sensors) within wireless sensor network applications. The research has explored ways by which intelligence can be integrated into wireless sensor networks in order to predict the optimal technique that is best suited to simple and complex scenario based on changing characteristics. The significant benefit of this includes being able to instantly select the best technique based on the current situations, as well as addressing new unseen scenarios, where optimal techniques can also be applied. Inclusive to these benefits is the fact that the network requires low maintenance, is self-organising and self-optimising.

The rest of this chapter discusses results obtained after various experiments as discussed in chapter 7 and relate these to the aim and objectives of the research. It also discusses various issues and challenges encountered during this process.

8.2 Results

The following results were obtained following this research:

1. The state of the art consists of the use of a DAT per scenario. This research was able to confirm that the existing approach was limited to selecting only one technique for the duration of the application scenario. It was discovered the selected technique did not perform optimally in all situations.

2. It was concluded that there was a strong relationship between the performance of a WSN, and the data aggregation technique applied across the network. This implies that given an application, with a set of requirements and objectives, there existed an opportunity to optimise the network by determining and applying optimal network setting.
3. Wireless sensor network applications (or scenarios) could be defined based on a set of characteristics (or features), the values of which directly affect the performance of the network while the application is running. This implied that the set of features could be used to model the state of a wireless sensor network application or scenario at any instant. This fact was important for this research because it supports the need to be able to evaluate the performance of the intelligent model, which was expected to be able to predict the optimal technique for a given scenario.
4. Given a scenario, and a set of techniques modelled within a simulation environment, it was proven that a single technique may not perform optimally across complex wireless sensor network scenarios. This implied that each scenario required a specific technique that performed optimally throughout the lifetime of the application. This fact was considered important for the research because the association of the data aggregation technique with a wireless sensor network was essential for the training of a machine learning model to predict a technique given new scenarios consistent with the method used in the state of the art, however, more dynamic.
5. The performance obtained from training a machine learning model with data on scenarios and the best techniques appropriate for such scenarios validated the fact that an intelligent model could be trained to learn these associations and to predict more accurately the right technique for a given scenario. This observation confirmed the objective of this research, which had the aim of designing and developing an intelligent model, which could dynamically manage a set of smart devices within a wireless sensor network application with the objective of optimising the performance. This also proves the hypothesis that there exist opportunities for the improvement the performance of wireless sensor networks by using an intelligent model which could predict optimal data aggregation technique in dynamic WSN scenarios.

8.3 Research Outcomes

The objectives of this research were approached systematically, and various conclusions were made. These are discussed below:

1. Research: after extensive study of literature, it was observed that there was a gap in the application of WSNs within various scenarios. The state of the art involved selecting a data aggregation technique, which was inherently infused in the hardware infrastructure of the network. Because wireless sensor networks are built for single unique applications, after deployment, WSNs are left unattended until their batteries run out. In this time, the networks are fixed on their data aggregation approach, independently of whether it performs optimally or not. Thus, there was a lack of a means to optimise the devices once deployed into the field. This provided justification to proceed with the research in order to investigate a new approach to optimise the use of WSNs within such dynamic environments.
2. Data: after extensive literature review, it was discovered that WSN applications required specific non-changeable techniques to determine their communication protocol. This fact was realised during the research and validated via experiments. It also formed the fundamental basis by which the new approach could be investigated. Thus, an appropriate set of features for this purpose were identified and data subsequently generated for a large set of single scenarios, which could be used to model various WSN complex scenarios.
3. Models: Based on data generated in objective 2, models were developed in a simulation environment to simulate the behaviour of WSN data aggregation techniques. Such models were used to simulate the performance of various techniques, which could then be used to assess the performance of the network. Based on this combination, various experiments were carried out to select best techniques given various complex scenarios. The resultant data was then collected as samples for training a machine learning model. This achievement implied that the appropriate data could be collected towards training a machine learning model to accurately predict optimal techniques for a given WSN scenario.
4. Intelligent Model: Based on objective 3, an intelligent machine learning model was designed and developed based on data generated in simulations involving the already developed data aggregation models. These were combined in various algorithms to develop a complete WSN network, where the required data could be generated for the machine learning model. The data collected at this stage was stored in a database for later dynamic querying to train the machine learning model. Afterwards, the model was built using a deep learning model.

5. Prototype: A working prototype of the model was developed to be able to evaluate its performance based on various use cases. It was shown that the model performs substantially better than the state-of-the-art approaches. Additionally, it provides an opportunity to explore more avenues of creating larger multipurpose systems for similar purposes.

8.4 Contributions

The following contributions where made based on the aim and objectives of the study:

1. Detection Needs Analysis Model
   A new methodology was developed for the management of wireless sensor network devices within numerous application scenarios which enabled the use of optimal algorithms for data aggregation based on the changing characteristics of the application. This was developed as a detection needs analysis model, which could be applied in future investigations in wireless sensor networks to identify the main components and attributes essential for the investigation.
2. Intelligent Dynamic and Adaptive Model
   A machine learning model was developed that was able to predict the best technique given various wireless sensor network application scenarios. The model was designed, tested, and evaluated to determine its accuracy, which was found to be impressive with respect to the state of the art. As discussed further in the recommendations, this model could serve in its current form as a viable recommendation solution, as well as providing the opportunity for enhancement by including more data on more techniques based on the aim of further studies.
3. Software Prototype, Framework and Source Code
   A framework that could be used to model new data aggregation techniques within the Network Simulator 3 simulation environment. This becomes important for the fact that there were limited sources for written code for the implementation of primary data aggregation techniques. the contributions consist of the following.
   a. Source code framework for the dynamic development of new WSN data aggregation techniques in NS3 simulation environment
   b. Dataset generated from experiments can be reused by other researchers
   c. Software prototype design for integration of the intelligent model for a web-based single query interface via RESTful API integration.

    d. Data generation approach based on NS3 simulator

4. Research Study's Experiments

   The conclusions of this research study are made based on the experimentation. These have provided further understanding about the development of an intelligent model to dynamically optimise wireless sensor networks. is The implementation and evaluation are also important contributions to the field in terms of exemplifying control automation and optimisation.

5. Applicability of Research Study's Outcome and Deliverables

   The outcome of this research contributes to the knowledge base, which supports that devices can be manufactured with sufficient intelligence to be able to self-optimise in any given wireless sensor-driven environment. This has far reaching impact within academia and industry, where the minimisation of resource consumption is an important factor to the system's operation.

8.5 Limitations and Recommendations

8.5.1 Limitations

This study presented an intelligent model for the improvement of the performance of WSN applications in terms of data processing. However, a few limitations are identified in the implementational of this study in the field. Some of these are discussed below:

1. Inclusion of more Techniques: the current study has only included only four techniques, which are LEACH, HEED, PEGASIS, and DBST. For the model to be considered as fit for use in the field, there is need to enable flexible training based on data collected from more scenarios and more DATs. Essentially, the intelligent model would need to be sufficiently dynamic to autonomously incorporate more data from more DATs, which perhaps have been dynamically detected by an external system. This ensures that the entire system remains ubiquitous and self-learning, requiring minimal maintenance.
2. Limitations of Current Sensors: currently produced sensor devices, including sensors already in operation, lack the infrastructure to collaboratively determine or recommend the best technique for the WSN network. In order for the results of this research to be useful in new applications, there is need for the inclusion of infrastructure to process data in order to determine the best technique in a method that

did not exist before and this is the innovative aspect of the proposed work herein. This could include that the sensors are able to host the recommendation model, or else, query a central source for the recommendation. This would also require preplanning and manufacturing design to incorporate hardware and software, which would be able to perform these tasks.

3. Maintenance of in-operation Sensors: irrespective of the application of the outcomes of this research, sensors deployed into operation would still require hardware maintenance. This consists of repair to damaged or comprised parts, including battery replacement. This is unavoidable since the devices would, at some stage, run out of power or be damaged based on their location. A new approach to address this problem needs to be investigated further to ensure that the sensors are completely autonomous and will no longer require physical maintenance once deployed into the field.

8.5.2 Future Recommendations

The following four recommendations have been identified:

1. Inclusion of More Techniques: The study has applied only four data aggregation techniques, which include LEACH, HEED, PEGASIS, and DBST. There is a potential to improve the performance of the model by enabling the model to become autonomous in the acquisition of new techniques. This would involve various steps, which include the determination of the objective of the technique, developing the technique model, data generation for the technique based on its best fit scenarios, and use of the data to improve the model. In this way, the model can improve itself as new complex scenarios arise. Such complex scenarios consist of more than one application running as the same time.
2. Integration with Physical Sensors: The research was implemented using a simulation environment, which has in-built models with capability to integrate with real physical sensors. This capability enables the data generated from the simulator to also include data from physical sensors. In this way, the model could be improved by enabling such integration, while using a stepwise improvement process to make the model more capable of managing physical devices with the appropriate internal infrastructure.
3. Use of Virtual Machines in Sensors: The study has used a simulation environment that enables models and other complex software to be infused into sensors, which are basically software as well. This makes it possible to improve the system by having sensors host virtual machines, which could run to host separate applications. This capability would

enable the sensors to run more than one data aggregation technique at a time and thus, more than one application at a time. This would improve the WSNs effectiveness and performance, while providing the flexibility for entertaining multiple scenarios.

4. Model Integration with Cloud Computing: As the research has focused on developing the model locally, that is without interaction with the internet, there is potential to have the model deployed into a cloud computing environment such as Amazon Web Services or Microsoft Azure. By developing a query interface to the model, this would allow the model's recommendations to be accessible via RESTful APIs to multiple consumers, both simulated and real. It would also enable the model to learn from new scenarios at greater scale.

8.6 Summary

This chapter has presented and discussed the results of the research study. These were aligned with the research aims and objectives under the research conclusions and were highlighted to have achieved the stated aim and objectives. The contributions were also discussed in terms of both academic relevance and contribution, as well as industrial innovation and impact to new wireless sensor-based systems. The limitations encountered and the those expected in the use of the model were also highlighted while future research and developments in the area of study was discussed.

In summary, this chapter concludes that the objectives and aims of this research have been met, while providing recommendations for the enhancement and improvement of the intelligent model.

Appendices

Appendix A

WSN Attributes selected based on literature on wireless sensor networks (referred to from section 4.6.1 and table 4.2)

No	Attribute	Sample Values	Data Type	Primary/ Derived	Static/Dynamic	Technique	Network	Scenario	Comments
1.	Node Count (A. Avokh and Mirjalily, 2010)	50, 100 no	Continuous	Primary	Dynamic	-	X	-	Dynamic based on active nodes
2.	Topology (Mantri et al., 2013; Wang et al., 2011)	Cluster, Tree	Ordinal	Primary	Static	X	-	-	-
3.	Homogeneity (Yi et al., 2007)	Homogenous, heterogenous	Binary	Primary	Static	X	-	-	-
4.	Field Size (Beydoun et al., 2009)	100 metres	Continuous	Primary	Static, Dynamic	-	X	X	Network nodes distribution or Scenario event perimeter
5.	Network Structure (Mamun, 2012)	Hierarchical, Flat	Ordinal	Primary	Static	X	-	-	-
6.	Node Mobility (Gnanasekaran and Francis, 2014)	True, False	Binary	Primary	Static	X	X	-	Technique standard requirement or network specification change
7.	Location Awareness (Al-Karaki and Kamal, 2004)	True, False	Binary	Primary	Static	X	-	-	Node has information about its location
8.	Network Awareness (Al-Karaki and Kamal, 2004)	True, False	Binary	Primary	Static	X	-	-	Node has information about other locations of all nodes

9.	Node residual energy (Chand et al., 2014)	10 Joules	Continuous	Primary	Dynamic	X	X	-	Technique performance metric or network specification update
10.	Event Type (Kim et al., 2016)	Detection / Monitoring	Ordinal	Primary	Static	-	-	X	Event type classification
11.	Node distribution (Bonomi and Milito, 2012)	Nodes / region	Continuous	Derived		-	X	-	-
12.	Node transmission Range (Mishra et al., 2017)	3 metres	Continuous	Primary	Dynamic	-	X	-	Nodes can modify their transmission range based on distance of destination node
13.	Simulation Rounds (Sasirekha and Swamynathan, 2015)	100	Continuous	Primary	Dynamic	X	X	-	-
14.	Maximum Distance between nodes (derived)	Max (distance btw nodes)	Continuous	Derived	Dynamic	-	X	-	-
15.	Average distance between nodes (derived)	Sum(distance) / node count	Continuous	Derived	Dynamic	-	X	-	-
16.	Transmission reliability (Villas et al., 2013)	Sampled by message sent	Continuous	Derived	Dynamic	X	-	-	Sampled message by message submitted to sink
17.	Energy distribution (Nie and Li, 2011)	Energy / area	Continuous	Derived	Dynamic	X	-	-	-
18.	Energy efficiency (Du et al., 2016)	Energy / packet	Continuous	Derived	Dynamic	X	-	-	-
19.	Throughput (Zhang and Guo, 2017)	Bytes / message	Continuous	Derived	Dynamic	X	-	-	-
20.	Latency (Bonomi and Milito, 2012)	m/sec	Continuous	Derived	Dynamic	X	-	-	-
21.	Stability (Gantassi et al., 2017)	Packets by active nodes / field size	Continuous	Derived	Dynamic	X	-	-	Packets sent by drop in active nodes, or change in field size
22.	Accuracy (Jesus et al., 2015)	bytes sensed / bytes sent	Continuous	Derived	Dynamic	X	-	-	-

23.	Bandwidth Utilization (Sahoo et al., 2017)	bytes	Continuous	Derived	Dynamic	X	-	-	-
24.	Computation time	seconds	Continuous	Derived	Dynamic	X	-	-	-
25.	Sampling Rate (reporting frequency) (derived)	count/secs	Continuous	Primary	Dynamic	-	X	X	Network setting or scenario event generated
26.	Control overhead (Al-Karaki and Kamal, 2004)	bytes / energy	Continuous	Derived	Dynamic	X	-	-	Bytes transmitted during topology construction by energy consumed
27.	Packet Delivery Ratio (Virmani et al., 2013)	Packets generated / submitted	Continuous	Derived	Dynamic	X	-	-	Packets generated by packets submitted to sink

Legend	
	Derived attributes used in computations
	Attribute classification under WSN entities
	Candidate attributes for performance indicators
X	There exists a link between the row/column elements
-	There is no connection between the row/column elements

Appendix B

Software Prototype Contents

The software prototype referred to severally in the study consists of the following items. Some of these were provided as attachments with the thesis document.

1. Software Prototype Software: a description of this was provided within the thesis. No extra content was made available for this
2. Wireless Sensor Data: this consists of the data generated from the NS3 simulator and was provided as an attachment with the thesis.
3. Source Code Framework: this consists of the code developed to model the WSN data aggregation techniques. The source code was provided as an attachment with the thesis.

Appendix C

Graph Theory

The following brief discussion introduces the concepts of Graph Theory as used in section 5.4 and other sections within the document (Rahman, M.S., 2017).

A graph can be defined to consist of a set of vertices, connected by a set of edges. When a graph theory is applied to a problem, vertices are usually used to represent objects, while edges are used to represent the relationship between two objects. Thus, a graph could be used to model the structure of the relationship between the two objects, including their communication method. In order words, graph theory can be applied to problems where objects and their relationships can be identified.

Formally, a graph G is a tuple (V, E), which consists of a finite set V of vertices, including a finite set E of edges. Since an edge could only be formed by the connection of two vertices, an edge could be an unordered pair of vertices.

The set of vertices in a graph G can be denoted as V(G), while the set of edges in G can be denoted as E(G). Two vertices u and v, could be said to be *adjacent* if an edge e = (u, v), and e is an edge in graph G. likewise, the edge e is said to be *incident* to u and v. The vertices u and v are also considered to be *neighbours* in G.

Graph theory is applied to numerous science and engineering problems and is used to systematically determine solutions to the problem addressed. Examples of problems it can be applied to include detecting the best path to deliver services to locations connected by road, gas supply network, wireless sensor networks, frequency assignment, floor planning, social networks, and bioinformatics (Rahman, M.S., 2017).

References

Abbas, N., Yu, F., 2018. A Traffic Congestion Control Algorithm for Wireless Multimedia Sensor Networks, in: 2018 IEEE SENSORS. Presented at the 2018 IEEE SENSORS, pp. 1–4. https://doi.org/10.1109/ICSENS.2018.8589923

Abidi, M.G., Ben Smida, M., Khalgui, M., Li, Z., Wu, N., 2017. Multi-agent-oriented solution for forecasting-based control strategy with load priority of microgrids in an island mode – Case study: Tunisian petroleum platform. Electr. Power Syst. Res. 152, 411–423. https://doi.org/10.1016/j.epsr.2017.07.013

Akyildiz, I.F., Su, W., Sankarasubramaniam, Y., Cayirci, E., 2002. A survey on sensor networks. IEEE Commun. Mag. 40, 102–114. https://doi.org/10.1109/MCOM.2002.1024422

Al Rasyid, M.U.H., Prasetyo, D., Nadhori, I.U., Alasiry, A.H., 2015. Mobile monitoring of muscular strain sensor based on Wireless Body Area Network, in: 2015 International Electronics Symposium (IES). Presented at the 2015 International Electronics Symposium (IES), pp. 284–287. https://doi.org/10.1109/ELECSYM.2015.7380856

Al-Doghman, F., Chaczko, Z., Jiang, J., 2017. A Review of Aggregation Algorithms for the Internet of Things, in: 2017 25th International Conference on Systems Engineering (ICSEng). Presented at the 2017 25th International Conference on Systems Engineering (ICSEng), pp. 480–487. https://doi.org/10.1109/ICSEng.2017.43

Alduais, N.A.M., Abdullah, I., Jamil, A., 2018. An Efficient Data Collection Algorithm for Wearable / Mobile Tracking System in IoT /WSN, in: 2018 Electrical Power, Electronics, Communications, Controls and Informatics Seminar (EECCIS). Presented at the 2018 Electrical Power, Electronics, Communications, Controls and Informatics Seminar (EECCIS), pp. 250–254. https://doi.org/10.1109/EECCIS.2018.8692815

Al-kahtani, M.S., Karim, L., 2018. Dynamic Data Aggregation Approach for Sensor-Based Big Data. Int. J. Adv. Comput. Sci. Appl. 9. https://doi.org/10.14569/IJACSA.2018.090710

Al-Karaki, J.N., Kamal, A.E., 2004. Routing techniques in wireless sensor networks: a survey. IEEE Wirel. Commun. 11, 6–28. https://doi.org/10.1109/MWC.2004.1368893

AlMansour, N., Alahmadi, S., 2018. Secure Ad Hoc On-Demand Distance Vector Routing Protocol in WSN, in: 2018 1st International Conference on Computer Applications Information Security (ICCAIS). Presented at the 2018 1st International Conference on

Computer Applications Information Security (ICCAIS), pp. 1–4. https://doi.org/10.1109/CAIS.2018.8441991

Almiani, M., AbuGhazleh, A., Al-Rahayfeh, A., Atiewi, S., Razaque, A., 2020. Deep recurrent neural network for IoT intrusion detection system. Simul. Model. Pract. Theory 101, 102031. https://doi.org/10.1016/j.simpat.2019.102031

Alnahdi, A., Liu, S.-H., 2017. Mobile Internet of Things (MIoT) and Its Applications for Smart Environments: A Positional Overview, in: 2017 IEEE International Congress on Internet of Things (ICIOT). Presented at the 2017 IEEE International Congress on Internet of Things (ICIOT), pp. 151–154. https://doi.org/10.1109/IEEE.ICIOT.2017.26

Alphonsa A., Ravi G., 2016. Earthquake early warning system by IOT using Wireless sensor networks, in: 2016 International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET). Presented at the 2016 International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET), pp. 1201–1205. https://doi.org/10.1109/WiSPNET.2016.7566327

Al-Tabbakh, S.M., 2017. Novel technique for data aggregation in wireless sensor networks, in: 2017 International Conference on Internet of Things, Embedded Systems and Communications (IINTEC). Presented at the 2017 International Conference on Internet of Things, Embedded Systems and Communications (IINTEC), pp. 1–8. https://doi.org/10.1109/IINTEC.2017.8325904

Anjali, Garg, A., Suhali, 2015. Distance Adaptive Threshold Sensitive Energy Efficient Sensor Network (DAPTEEN) protocol in WSN, in: 2015 International Conference on Signal Processing, Computing and Control (ISPCC). Presented at the 2015 International Conference on Signal Processing, Computing and Control (ISPCC), pp. 114–119. https://doi.org/10.1109/ISPCC.2015.7375008

Antonopoulos, C., Touliatos, G., Voros, N., Koubias, S., 2018. Enhanced WSN simulation accuracy through novel hardware in the loop approach, in: 2018 IEEE International Conference on Industrial Technology (ICIT). Presented at the 2018 IEEE International Conference on Industrial Technology (ICIT), pp. 1946–1951. https://doi.org/10.1109/ICIT.2018.8352484

Ari, D., ÇIBUK, M., AGGUN, F., 2018. The Comparison of Energy Consumption of Different Topologies in Multi-hop Wireless Sensor Networks | Request PDF [WWW Document]. ResearchGate. URL https://www.researchgate.net/publication/331422362_The_Comparison_of_Energy_Consumption_of_Different_Topologies_in_Multi-hop_Wireless_Sensor_Networks (accessed 3.12.19).

Arshad, M., Alsalem, M., Siddqui, F.A., Kamel, N., Saad, N.M., 2012. Efficient cluster head selection scheme in Mobile Data Collector based routing protocol, in: 2012 4th International Conference on Intelligent and Advanced Systems (ICIAS2012). Presented at the 2012 4th International Conference on Intelligent and Advanced Systems (ICIAS2012), pp. 280–284. https://doi.org/10.1109/ICIAS.2012.6306202

Assis, L.F.F.G. de, Behnck, L.P., Doering, D., Freitas, E.P. de, Pereira, C.E., Horita, F.E.A., Ueyama, J., Albuquerque, J.P. de, 2016. Dynamic Sensor Management: Extending Sensor Web for Near Real-Time Mobile Sensor Integration in Dynamic Scenarios, in: 2016 IEEE 30th International Conference on Advanced Information Networking and Applications (AINA). Presented at the 2016 IEEE 30th International Conference on Advanced Information Networking and Applications (AINA), pp. 303–310. https://doi.org/10.1109/AINA.2016.100

Atat, R., Liu, L., Wu, J., Li, G., Ye, C., Yang, Y., 2018. Big Data Meet Cyber-Physical Systems: A Panoramic Survey. IEEE Access 6, 73603–73636. https://doi.org/10.1109/ACCESS.2018.2878681

Atlam, H.F., Walters, R.J., Wills, G.B., 2018. Fog Computing and the Internet of Things: A Review. Big Data Cogn. Comput. 2, 10. https://doi.org/10.3390/bdcc2020010

Avokh, Avid, Mirjalily, G., 2010. Dynamic Balanced Spanning Tree (DBST) for data aggregation in Wireless Sensor Networks, in: 2010 5th International Symposium on Telecommunications. Presented at the 2010 5th International Symposium on Telecommunications, pp. 391–396. https://doi.org/10.1109/ISTEL.2010.5734058

Avokh, A., Mirjalily, G., 2010. Dynamic Balanced Spanning Tree (DBST) for data aggregation in Wireless Sensor Networks, in: 2010 5th International Symposium on Telecommunications. Presented at the 2010 5th International Symposium on Telecommunications, pp. 391–396. https://doi.org/10.1109/ISTEL.2010.5734058

Ayaz, M., Ammad-uddin, M., Baig, I., Aggoune, el-H.M., 2018. Wireless Sensor's Civil Applications, Prototypes, and Future Integration Possibilities: A Review. IEEE Sens. J. 18, 4–30. https://doi.org/10.1109/JSEN.2017.2766364

Bakas, S., Uttamchandani, D., Toshiyoshi, H., Bauer, R., 2019. A miniaturised light-sheet microscopy system using MEMS micromirror control, in: 2019 International

Conference on Optical MEMS and Nanophotonics (OMN). Presented at the 2019 International Conference on Optical MEMS and Nanophotonics (OMN), pp. 176–177. https://doi.org/10.1109/OMN.2019.8925053

Baldovino, R.G., Valenzuela, I.C., Dadios, E.P., 2018. Implementation of a Low-power Wireless Sensor Network for Smart Farm Applications, in: 2018 IEEE 10th International Conference on Humanoid, Nanotechnology, Information Technology,Communication and Control, Environment and Management (HNICEM). Presented at the 2018 IEEE 10th International Conference on Humanoid, Nanotechnology, Information Technology,Communication and Control, Environment and Management (HNICEM), pp. 1–5. https://doi.org/10.1109/HNICEM.2018.8666262

Bangotra, D.K., Singh, Y., Selwal, A., 2018. Machine Learning in Wireless Sensor Networks: Challenges and Opportunities, in: 2018 Fifth International Conference on Parallel, Distributed and Grid Computing (PDGC). Presented at the 2018 Fifth International Conference on Parallel, Distributed and Grid Computing (PDGC), pp. 534–539. https://doi.org/10.1109/PDGC.2018.8745845

Behal, A., Vyas, A., Pal, S., 2019. Local Traffic Optimization by Exploiting the Mobility of Smart Devices, in: 2019 11th International Conference on Communication Systems Networks (COMSNETS). Presented at the 2019 11th International Conference on Communication Systems Networks (COMSNETS), pp. 690–694. https://doi.org/10.1109/COMSNETS.2019.8711420

Benjelloun, F.-Z., Lahcen, A.A., Belfkih, S., 2015. An overview of big data opportunities, applications and tools, in: 2015 Intelligent Systems and Computer Vision (ISCV). Presented at the 2015 Intelligent Systems and Computer Vision (ISCV), pp. 1–6. https://doi.org/10.1109/ISACV.2015.7105553

Beydoun, K., Felea, V., Guyennet, H., 2009. Wireless Sensor Network Infrastructure: Construction and Evaluation, in: 2009 Fifth International Conference on Wireless and Mobile Communications. Presented at the 2009 Fifth International Conference on Wireless and Mobile Communications, pp. 279–284. https://doi.org/10.1109/ICWMC.2009.53

Bongale, A.M., Swarup, A., Shivam, S., 2017. EiP-LEACH: Energy influenced probability based LEACH protocol for Wireless Sensor Network, in: 2017 International Conference on Emerging Trends Innovation in ICT (ICEI). Presented at the 2017

International Conference on Emerging Trends Innovation in ICT (ICEI), pp. 77–81. https://doi.org/10.1109/ETIICT.2017.7977014

Bonomi, F., Milito, R., 2012. Fog Computing and its Role in the Internet of Things. Proc. MCC Workshop Mob. Cloud Comput. https://doi.org/10.1145/2342509.2342513

Boubiche, S., Boubiche, D.E., Bilami, A., Toral-Cruz, H., 2018. Big Data Challenges and Data Aggregation Strategies in Wireless Sensor Networks. IEEE Access 6, 20558–20571. https://doi.org/10.1109/ACCESS.2018.2821445

Burrell, G. and Morgan, G. (1982) Sociological Paradigms and Organisational Analysis. London: Heinemann.

Chand, S., Singh, S., Kumar, B., 2014. Heterogeneous HEED Protocol for Wireless Sensor Networks. Wirel. Pers. Commun. 77, 2117–2139. https://doi.org/10.1007/s11277-014-1629-y

Chatterjee, P., Armentano, R.L., Cymberknop, L.J., 2017. Internet of Things and Decision Support System for eHealth - Applied to Cardiometabolic Diseases, in: 2017 International Conference on Machine Learning and Data Science (MLDS). Presented at the 2017 International Conference on Machine Learning and Data Science (MLDS), pp. 75–79. https://doi.org/10.1109/MLDS.2017.22

Chen, Y., Wang, N.-C., Chen, C.-L., Lin, Y.-C., 2011. A Coverage Algorithm to Improve the Performance of PEGASIS in Wireless Sensor Networks. pp. 123–127. https://doi.org/10.1109/SNPD.2011.11

Chiang, M., Zhang, T., 2016. Fog and IoT: An Overview of Research Opportunities. IEEE Internet Things J. 3, 854–864. https://doi.org/10.1109/JIOT.2016.2584538

Chniter, H., Li, Y., Khalgui, M., Koubaa, A., Li, Z., Jarray, F., 2018. Multi-Agent Adaptive Architecture for Flexible Distributed Real-Time Systems. IEEE Access 6, 23152–23171. https://doi.org/10.1109/ACCESS.2018.2825023

Dahda, N., Sindhwani, M., Singh, C., 2018. Energy Efficient Hybrid Technique Based on Dynamic Clustering in Wireless Sensor Network, in: 2018 International Conference on Intelligent Circuits and Systems (ICICS). Presented at the 2018 International Conference on Intelligent Circuits and Systems (ICICS), pp. 229–234. https://doi.org/10.1109/ICICS.2018.00055

Daiya, V., krishnan, T.S.S., Ebenezer, J., Madhusoodanan, K., SatyaMurty, S.A.V., Rao, B., 2016. Dynamic architecture for Wireless Sensor Network-implementation analysis, in: 2016 International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET). Presented at the 2016 International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET), pp. 1206–1211. https://doi.org/10.1109/WiSPNET.2016.7566328

Darabkh, K.A., Al-Rawashdeh, W.S., Hawa, M., Saifan, R., Khalifeh, A.F., 2017. A novel clustering protocol for wireless sensor networks, in: 2017 International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET). Presented at the 2017 International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET), pp. 435–438. https://doi.org/10.1109/WiSPNET.2017.8299793

Deloitte, I., 2018. TechTrends 2018 FINAL.

Dhand, G., Tyagi, S.S., 2016. Data Aggregation Techniques in WSN:Survey. Procedia Comput. Sci., 2nd International Conference on Intelligent Computing, Communication & Convergence, ICCC 2016, 24-25 January 2016, Bhubaneswar, Odisha, India 92, 378–384. https://doi.org/10.1016/j.procs.2016.07.393

Diro, A.A., Chilamkurti, N., 2018. Distributed attack detection scheme using deep learning approach for Internet of Things. Future Gener. Comput. Syst. 82, 761–768. https://doi.org/10.1016/j.future.2017.08.043

Djedouboum, A.C., Ari, A.A.A., Guéroui, A.M., Mohamadou, A., Aliouat, Z., 2018. Big Data Collection in Large-Scale Wireless Sensor Networks, in: Sensors. https://doi.org/10.3390/s18124474

Du, T., Qu, S., Liu, K., Xu, J., Cao, Y., 2016. An Efficient Data Aggregation Algorithm for WSNs Based on Dynamic Message List. Procedia Comput. Sci. 83, 98–106. https://doi.org/10.1016/j.procs.2016.04.104

Galappaththi, H.R., Weerasuriya, G.T., 2018. Survey on Wireless Sensor Networks (WSNs) Implemented for Environmental Sensing, in: 2018 3rd International Conference on Information Technology Research (ICITR). Presented at the 2018 3rd International Conference on Information Technology Research (ICITR), pp. 1–6. https://doi.org/10.1109/ICITR.2018.8736151

Ganguly, P., Nasipuri, M., Dutta, S., 2019. Challenges of the Existing Security Measures Deployed in the Smart Grid Framework, in: 2019 IEEE 7th International Conference on Smart Energy Grid Engineering (SEGE). Presented at the 2019 IEEE 7th International Conference on Smart Energy Grid Engineering (SEGE), pp. 1–5. https://doi.org/10.1109/SEGE.2019.8859917

Gantassi, R., Ben Yagouta, A., Ben Gouissem, B., 2017. Improvement of the LEACH protocol in load balancing and energy-association conservation, in: 2017 International Conference on Internet of Things, Embedded Systems and Communications (IINTEC). Presented at the 2017 International Conference on Internet of Things, Embedded Systems and Communications (IINTEC), pp. 53–59. https://doi.org/10.1109/IINTEC.2017.8325913

Gasmi, M., Mosbahi, O., Khalgui, M., Gomes, L., Li, Z., 2018a. R-Node: New Pipelined Approach for an Effective Reconfigurable Wireless Sensor Node. IEEE Trans. Syst. Man Cybern. Syst. 48, 892–905. https://doi.org/10.1109/TSMC.2016.2625817

Gasmi, M., Mosbahi, O., Khalgui, M., Gomes, L., Li, Z., 2018b. Performance Optimization of Reconfigurable Real-Time Wireless Sensor Networks. IEEE Trans. Syst. Man Cybern. Syst. 1–15. https://doi.org/10.1109/TSMC.2018.2824900

Ghai, S., Katiyar, V.K., 2016. Data Aggregation to Improve Energy Efficiency in Wireless Sensor Networks. Int. J. Innov. Res. Inf. Secur. 3, 4.

Ghosh, A., Maeder, A., Baker, M., Chandramouli, D., 2019. 5G Evolution: A View on 5G Cellular Technology Beyond 3GPP Release 15. IEEE Access 7, 127639–127651. https://doi.org/10.1109/ACCESS.2019.2939938

Gnanasekaran, T., Francis, S.A.J., 2014. Comparative analysis on routing protocols in Wireless Sensor Networks, in: 2014 International Conference on Computer Communication and Informatics. Presented at the 2014 International Conference on Computer Communication and Informatics, pp. 1–6. https://doi.org/10.1109/ICCCI.2014.6921777

Grichi, H., Mosbahi, O., Khalgui, M., Li, Z., 2017. RWiN: New Methodology for the Development of Reconfigurable WSN. IEEE Trans. Autom. Sci. Eng. 14, 109–125. https://doi.org/10.1109/TASE.2016.2608918

Gupta, G.S., Quan, V.M., 2018. Multi-sensor integrated system for wireless monitoring of greenhouse environment, in: 2018 IEEE Sensors Applications Symposium (SAS). Presented at the 2018 IEEE Sensors Applications Symposium (SAS), pp. 1–6. https://doi.org/10.1109/SAS.2018.8336723

Gupta, M., Saraswat, L., 2014. Energy aware data collection in wireless sensor network using chain based PEGASIS, in: International Conference on Recent Advances and Innovations in Engineering (ICRAIE-2014). Presented at the International Conference on Recent Advances and Innovations in Engineering (ICRAIE-2014), pp. 1–5. https://doi.org/10.1109/ICRAIE.2014.6909182

Handy, M.J., Haase, M., Timmermann, D., 2002. Low energy adaptive clustering hierarchy with deterministic cluster-head selection, in: 4th International Workshop on Mobile and Wireless Communications Network. Presented at the 4th International Workshop on Mobile and Wireless Communications Network, pp. 368–372. https://doi.org/10.1109/MWCN.2002.1045790

Hao Qi, Fubao Wang, Bingxiang Wang, 2011. Energy aware adaptive directed diffusion algorithm of wireless sensor networks, in: 2011 International Conference on Computer Science and Service System (CSSS). Presented at the 2011 International Conference on Computer Science and Service System (CSSS), pp. 385–388. https://doi.org/10.1109/CSSS.2011.5972028

Harb, H., Makhoul, A., Laiymani, D., Jaber, A., Tawil, R., 2014a. K-means based clustering approach for data aggregation in periodic sensor networks, in: 2014 IEEE 10th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob). Presented at the 2014 IEEE 10th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob), pp. 434–441. https://doi.org/10.1109/WiMOB.2014.6962207

Harb, H., Makhoul, A., Tawil, R., Jaber, A., 2014b. Energy-efficient data aggregation and transfer in periodic sensor networks. IET Wirel. Sens. Syst. 4, 149–158. https://doi.org/10.1049/iet-wss.2014.0068

Hasan, M., Islam, Md.M., Zarif, M.I.I., Hashem, M.M.A., 2019. Attack and anomaly detection in IoT sensors in IoT sites using machine learning approaches. Internet Things 7, 100059. https://doi.org/10.1016/j.iot.2019.100059

Heinzelman, W.R., Chandrakasan, A., Balakrishnan, H., 2000. Energy-efficient communication protocol for wireless microsensor networks, in: Proceedings of the 33rd Annual Hawaii International Conference on System Sciences. Presented at the Proceedings of the 33rd Annual Hawaii International Conference on System Sciences, p. 10 pp. vol.2-. https://doi.org/10.1109/HICSS.2000.926982

Hiteshreddy, O., Singh, P., Chahuan, S., 2015. A Review on Cluster Based Data Aggregation Protocols in Wireless Sensor Network. Int. J. Comput. Sci. Eng. 3, 9.

Hooda, N., Bawa, S., Rana, P.S., 2018. MCTOPE Ensemble Machine Learning Framework: A Case Study of Routing Protocol Prediction, in: 2018 IEEE 3rd International

Conference on Computing, Communication and Security (ICCCS). Presented at the 2018 IEEE 3rd International Conference on Computing, Communication and Security (ICCCS), pp. 92–99. https://doi.org/10.1109/CCCS.2018.8586811

Huang, Z., Zheng, J., 2012. A Slepian-Wolf coding based energy-efficient clustering algorithm for data aggregation in wireless sensor networks, in: 2012 IEEE International Conference on Communications (ICC). Presented at the 2012 IEEE International Conference on Communications (ICC), pp. 198–202. https://doi.org/10.1109/ICC.2012.6363998

Intanagonwiwat, C., Govindan, R., Estrin, D., Heidemann, J., Silva, F., 2003. Directed diffusion for wireless sensor networking. IEEEACM Trans. Netw. 11, 2–16. https://doi.org/10.1109/TNET.2002.808417

Jafri, M., Javaid, N., Javaid, A., Khan, Z., 2013. Maximizing the Lifetime of Multi-chain PEGASIS using Sink Mobility. World Appl. Sci. J. 21. https://doi.org/10.5829/idosi.wasj.2013.21.9.175

Jesus, P., Baquero, C., Almeida, P.S., 2015. A Survey of Distributed Data Aggregation Algorithms. IEEE Commun. Surv. Tutor. 17, 381–404. https://doi.org/10.1109/COMST.2014.2354398

Jiang, W., Gu, C., Wu, J., 2017. A Dynamically Reconfigurable Wireless Sensor Network Testbed for Multiple Routing Protocols [WWW Document]. Wirel. Commun. Mob. Comput. https://doi.org/10.1155/2017/1594270

Johnson, P. and Clark, M. (2006) 'Mapping the terrain: an overview of business and management research methodologies', in P. Johnson and M. Clark. (eds) Business and Management Research Methodologies. London: Sage.Juneja, A., Das, N.N., 2019. Big Data Quality Framework: Pre-Processing Data in Weather Monitoring Application, in: 2019 International Conference on Machine Learning, Big Data, Cloud and Parallel Computing (COMITCon). Presented at the 2019 International Conference on Machine Learning, Big Data, Cloud and Parallel Computing (COMITCon), pp. 559–563. https://doi.org/10.1109/COMITCon.2019.8862267

Jung, J., Kang, D., Choi, J., Min, O., 2016. Near field distance awareness algorithm using bluetooth for device sociality service, in: 2016 18th International Conference on Advanced Communication Technology (ICACT). Presented at the 2016 18th International Conference on Advanced Communication Technology (ICACT), pp. 140–143. https://doi.org/10.1109/ICACT.2016.7423302

Jyothi, R., Cholli, N.G., 2019. New Approach to Secure Cluster Heads in Wireless Sensor Networks, in: 2019 5th International Conference on Advanced Computing Communication Systems (ICACCS). Presented at the 2019 5th International Conference on Advanced Computing Communication Systems (ICACCS), pp. 1097–1101. https://doi.org/10.1109/ICACCS.2019.8728380

K., S., Vaidehi, V., 2018. Clustering and Data Aggregation in Wireless Sensor Networks Using Machine Learning Algorithms, in: 2018 International Conference on Recent Trends in Advance Computing (ICRTAC). Presented at the 2018 International Conference on Recent Trends in Advance Computing (ICRTAC), pp. 109–115. https://doi.org/10.1109/ICRTAC.2018.8679318

Kang, B., Park, C., Choo, H., 2019. A Location Aware Fast PMIPv6 for Low Latency Wireless Sensor Networks. IEEE Sens. J. 19, 9456–9467. https://doi.org/10.1109/JSEN.2019.2925637

Karoui, O., Khalgui, M., Koubâa, A., Guerfala, E., Li, Z., Tovar, E., 2017. Dual mode for vehicular platoon safety: Simulation and formal verification. Inf. Sci. 402, 216–232. https://doi.org/10.1016/j.ins.2017.03.016

Khan, Z.A., Samad, A., 2017. A Study of Machine Learning in Wireless Sensor Network. Int. J. Comput. Netw. Appl. 4. https://doi.org/10.22247/ijcna/2017/49122

Kim, J.-C., Jung, H., Kim, S., Chung, K., 2016. Slope Based Intelligent 3D Disaster Simulation Using Physics Engine. Wirel. Pers. Commun. 86, 183–199. https://doi.org/10.1007/s11277-015-2788-1

Kulik, J., Heinzelman, W., Balakrishnan, H., 2002. Negotiation-Based Protocols for Disseminating Information in Wireless Sensor Networks. Wirel. Netw. 8, 169–185. https://doi.org/10.1023/A:1013715909417

Kulshrestha, J., Mishra, M.K., 2017. DPEGASIS: Distributed PEGASIS for chain construction by the nodes in the network or in a zone without having global network topology information, in: 2017 International Conference on Multimedia, Signal Processing and Communication Technologies (IMPACT). Presented at the 2017 International Conference on Multimedia, Signal Processing and Communication Technologies (IMPACT), pp. 13–17. https://doi.org/10.1109/MSPCT.2017.8363964

Kumar, D., Aseri, T.C., Patel, R.B., 2009. EEHC: Energy efficient heterogeneous clustered scheme for wireless sensor networks. Comput Commun. https://doi.org/10.1016/j.comcom.2008.11.025

Kumar Dwivedi, R., Pandey, S., Kumar, R., 2018. A Study on Machine Learning Approaches for Outlier Detection in Wireless Sensor Network, in: 2018 8th International Conference on Cloud Computing, Data Science Engineering (Confluence). Presented at the 2018 8th International Conference on Cloud Computing, Data Science Engineering (Confluence), pp. 189–192. https://doi.org/10.1109/CONFLUENCE.2018.8442992

Kuncoro, cb, Falahuddin, A., 2014. Design Concept of Dynamic-Adaptive Reconfigurable Wireless Sensor Node (DARWiSeN). TELKOMNIKA Telecommun. Comput. Electron. Control 12, 695. https://doi.org/10.12928/v12i3.98

Li, X., Huang, H., Sun, Y., 2016. DriTri: An in-vehicle wireless sensor network platform for daily health monitoring, in: 2016 IEEE SENSORS. Presented at the 2016 IEEE SENSORS, pp. 1–3. https://doi.org/10.1109/ICSENS.2016.7808894

Lindsey, S., Raghavendra, C.S., 2002. PEGASIS: Power-efficient gathering in sensor information systems, in: Proceedings, IEEE Aerospace Conference. Presented at the Proceedings, IEEE Aerospace Conference, pp. 3-1125-3–1130 vol.3. https://doi.org/10.1109/AERO.2002.1035242

Luong, N.C., Wang, P., Niyato, D., Wen, Y., Han, Z., 2017. Resource Management in Cloud Networking Using Economic Analysis and Pricing Models: A Survey. IEEE Commun. Surv. Tutor. 19, 954–1001. https://doi.org/10.1109/COMST.2017.2647981

Luwei Jing, Feng Liu, Yuling Li, 2011. Energy saving routing algorithm based on SPIN protocol in WSN, in: 2011 International Conference on Image Analysis and Signal Processing. Presented at the 2011 International Conference on Image Analysis and Signal Processing, pp. 416–419. https://doi.org/10.1109/IASP.2011.6109074

Ma, Z., Xiao, M., Xiao, Y., Pang, Z., Poor, H.V., Vucetic, B., 2019. High-Reliability and Low-Latency Wireless Communication for Internet of Things: Challenges, Fundamentals, and Enabling Technologies. IEEE Internet Things J. 6, 7946–7970. https://doi.org/10.1109/JIOT.2019.2907245

Mamun, Q., 2012. A Qualitative Comparison of Different Logical Topologies for Wireless Sensor Networks. Sensors 12, 14887–14913. https://doi.org/10.3390/s121114887

Mantri, D., Prasad, N.R., Prasad, R., 2013. Grouping of clusters for efficient data aggregation (GCEDA) in wireless sensor network, in: 2013 3rd IEEE International Advance Computing Conference (IACC). Presented at the 2013 3rd IEEE International

Advance Computing Conference (IACC), pp. 132–137. https://doi.org/10.1109/IAdCC.2013.6514208

Mathapati, B.S., Patil, S.R., Mytri, V.D., 2012. Energy Efficient Reliable Data Aggregation Technique for Wireless Sensor Networks, in: 2012 International Conference on Computing Sciences. Presented at the 2012 International Conference on Computing Sciences, pp. 153–158. https://doi.org/10.1109/ICCS.2012.34

Meena, S.S., Manikandan, J., 2017. Study and evaluation of different topologies in wireless sensor network, in: 2017 International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET). Presented at the 2017 International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET), IEEE, Chennai, pp. 107–111. https://doi.org/10.1109/WiSPNET.2017.8299729

Mehta, D., Saxena, S., 2018. A Comparative Analysis of Energy Efficient Hierarchical Routing Protocols for Wireless Sensor Networks, in: 2018 4th International Conference on Computing Sciences (ICCS). Presented at the 2018 4th International Conference on Computing Sciences (ICCS), pp. 53–58. https://doi.org/10.1109/ICCS.2018.00015

Meskina, S.B., Doggaz, N., Khalgui, M., Li, Z., 2018. Reconfiguration-based methodology for improving recovery performance of faults in smart grids. Inf. Sci. 454–455, 73–95. https://doi.org/10.1016/j.ins.2018.04.010

Mishra, S., Bano, R., Kumar, S., Dixit, V., 2017. A literature survey on routing protocol in wireless sensor network, in: 2017 International Conference on Innovations in Information, Embedded and Communication Systems (ICIIECS). Presented at the 2017 International Conference on Innovations in Information, Embedded and Communication Systems (ICIIECS), pp. 1–4. https://doi.org/10.1109/ICIIECS.2017.8276150

Mohanty, P., Kabat, M.R., 2016. Energy efficient structure-free data aggregation and delivery in WSN. Egypt. Inform. J. 17, 273–284. https://doi.org/10.1016/j.eij.2016.01.002

Nie, P., Li, B., 2011. A cluster-based data aggregation architecture in WSN for Structural Health Monitoring, in: 2011 7th International Wireless Communications and Mobile Computing Conference. Presented at the 2011 7th International Wireless Communications and Mobile Computing Conference, pp. 546–552. https://doi.org/10.1109/IWCMC.2011.5982592

Nishikawa, Y., Sasamura, T., Ishizuka, Y., Sugimoto, S., Iwasaki, S., Wang, H., Fujishima, T., Fujimoto, T., Yamashita, K., Suzuki, T., Kurihara, K., 2018. Design of stable wireless sensor network for slope monitoring, in: 2018 IEEE Topical Conference on Wireless Sensors and Sensor Networks (WiSNet). Presented at the 2018 IEEE Topical Conference on Wireless Sensors and Sensor Networks (WiSNet), pp. 8–11. https://doi.org/10.1109/WISNET.2018.8311550

NS3 Manual, 2020. ns-3, Ns-3. URL https://www.nsnam.org/docs/manual/html/index.html (accessed 4.27.20).

Omosebi, O, Bessis, N, Korkontzelos, Y, Pournaras, E, Sun, Q & Sotiriadis, S 2018, Dynamic Scenario-based Selection of Data Aggregation Techniques.. , International Academic, Research, and Industry Association (IARIA) Data Analytics 2018, Athens, Greece, 18/11/18, pp. 27-32 Otoum, S., Kantarci, B., Mouftah, H.T., 2019. On the Feasibility of Deep Learning in Sensor Network Intrusion Detection. IEEE Netw. Lett. 1, 68–71. https://doi.org/10.1109/LNET.2019.2901792

Ozbey, B., Kurc, O., Demir, H.V., Erturk, V.B., Altintas, A., 2018. Wireless Monitoring of a Structural Beam to be Used for Post-Earthquake Damage Assessment, in: 2018 2nd URSI Atlantic Radio Science Meeting (AT-RASC). Presented at the 2018 2nd URSI Atlantic Radio Science Meeting (AT-RASC), pp. 1–3. https://doi.org/10.23919/URSI-AT-RASC.2018.8471313

Panchal, A., Singh, R.K., 2018. Energy Efficient Cluster Head Selection with Adaptive Threshold in WSN, in: 2018 International Symposium on Intelligent Signal Processing and Communication Systems (ISPACS). Presented at the 2018 International Symposium on Intelligent Signal Processing and Communication Systems (ISPACS), pp. 127–130. https://doi.org/10.1109/ISPACS.2018.8923283

Pant, S., Kumar, R., Singh, A., 2017. Adaptive sink transmission and relocation to extend the network lifetime of wireless sensor network, in: 2017 3rd International Conference on Advances in Computing,Communication Automation (ICACCA) (Fall). Presented at the 2017 3rd International Conference on Advances in Computing,Communication Automation (ICACCA) (Fall), pp. 1–4. https://doi.org/10.1109/ICACCAF.2017.8344693

Patil, P., Kulkarni, U., 2013. SVM based data redundancy elimination for data aggregation in Wireless Sensor Networks, in: 2013 International Conference on Advances in Computing, Communications and Informatics (ICACCI). Presented at the 2013

International Conference on Advances in Computing, Communications and Informatics (ICACCI), pp. 1309–1316. https://doi.org/10.1109/ICACCI.2013.6637367

Patil, P.K., Patil, S.R., 2017. Structural health monitoring system using WSN for bridges, in: 2017 International Conference on Intelligent Computing and Control Systems (ICICCS). Presented at the 2017 International Conference on Intelligent Computing and Control Systems (ICICCS), pp. 371–375. https://doi.org/10.1109/ICCONS.2017.8250746

Pievanelli, E., Plesca, A., Stefanelli, R., Trinchero, D., 2013. Dynamic wireless sensor networks for real time safeguard of workers exposed to physical agents in constructions sites, in: 2013 IEEE Topical Conference on Wireless Sensors and Sensor Networks (WiSNet). Presented at the 2013 IEEE Topical Conference on Wireless Sensors and Sensor Networks (WiSNet), pp. 55–57. https://doi.org/10.1109/WiSNet.2013.6488632

Pillai, M.R., Jain, R.B., 2018. Application Specific Node Deployment in WSN, in: 2018 IEEE Global Conference on Wireless Computing and Networking (GCWCN). Presented at the 2018 IEEE Global Conference on Wireless Computing and Networking (GCWCN), pp. 168–172. https://doi.org/10.1109/GCWCN.2018.8668617

Popov, O., Kuzminykh, I., 2018. Analysis of methods for reducing topology in wireless sensor networks, in: 2018 14th International Conference on Advanced Trends in Radioelecrtronics, Telecommunications and Computer Engineering (TCSET). Presented at the 2018 14th International Conference on Advanced Trends in Radioelecrtronics, Telecommunications and Computer Engineering (TCSET), pp. 529–532. https://doi.org/10.1109/TCSET.2018.8336257

Postránecký, M., Svítek, M., 2017. Smart city near to 4.0 — an adoption of industry 4.0 conceptual model, in: 2017 Smart City Symposium Prague (SCSP). Presented at the 2017 Smart City Symposium Prague (SCSP), pp. 1–5. https://doi.org/10.1109/SCSP.2017.7973870

Pournaras, E., Nikolić, J., 2017. On-demand self-adaptive data analytics in large-scale decentralized networks, in: 2017 IEEE 16th International Symposium on Network Computing and Applications (NCA). Presented at the 2017 IEEE 16th International Symposium on Network Computing and Applications (NCA), pp. 1–10. https://doi.org/10.1109/NCA.2017.8171353

Prabha, R., Ramesh, M.V., Rangan, V.P., 2018. Building Optimal Topologies for Real-Time Wireless Sensor Networks, in: 2018 International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET). Presented at the 2018 International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET), pp. 1–6. https://doi.org/10.1109/WiSPNET.2018.8538721

Praveen Kumar, D., Amgoth, T., Annavarapu, C.S.R., 2019. Machine learning algorithms for wireless sensor networks: A survey. Inf. Fusion 49, 1–25. https://doi.org/10.1016/j.inffus.2018.09.013

Punniamoorthy, D., Kamadal, V.S., Srujana Yadav, B., Reddy, V.S., 2018. Wireless Sensor Networks for Effective Environmental Tracking System Using IoT and Sensors, in: 2018 2nd International Conference on 2018 2nd International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC)I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC). Presented at the 2018 2nd International Conference on 2018 2nd International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC)I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC), pp. 66–69. https://doi.org/10.1109/I-SMAC.2018.8653672

Puvaneshwari S, Vijayashaarathi S, 2016. Efficient Monitoring system for cardiac patients using Wireless Sensor Networks (WSN), in: 2016 International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET). Presented at the 2016 International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET), pp. 1558–1561. https://doi.org/10.1109/WiSPNET.2016.7566398

Qing, li, Zhu, Q., Wang, M., 2006. Design of a Distributed Energy-efficient Clustering Algorithm for Heterogeneous Wireless Sensor Networks. Comput. Commun. 29, 2230–2237. https://doi.org/10.1016/j.comcom.2006.02.017

Raghunandan, G.H., Rani, A.S., Nanditha, S.Y., Swathi, G., 2017. Hierarchical agglomerative clustering based routing algorithm for overall efficiency of wireless sensor network, in: 2017 International Conference on Intelligent Computing, Instrumentation and Control Technologies (ICICICT). Presented at the 2017 International Conference on Intelligent Computing, Instrumentation and Control Technologies (ICICICT), pp. 1290–1293. https://doi.org/10.1109/ICICICT1.2017.8342755

Rahman, H., Ahmed, N., Hussain, I., 2016. Comparison of data aggregation techniques in Internet of Things (IoT), in: 2016 International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET). Presented at the 2016 International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET), pp. 1296–1300. https://doi.org/10.1109/WiSPNET.2016.7566346

Rahman, M.S., 2017. Basic Graph Theory, 1st ed. 2017. ed, Undergraduate Topics in Computer Science. Springer International Publishing: Imprint: Springer, Cham. https://doi.org/10.1007/978-3-319-49475-3

Rajagopalan, R., Varshney, P.K., 2006. Data-aggregation techniques in sensor networks: A survey. IEEE Commun. Surv. Tutor. 8, 48–63. https://doi.org/10.1109/COMST.2006.283821

Randhawa, S., Jain, S., 2017a. Data Aggregation in Wireless Sensor Networks: Previous Research, Current Status and Future Directions. Wirel Commun 97, 3355–3425. https://doi.org/10.1007/s11277-017-4674-5

Randhawa, S., Jain, S., 2017b. DAHDA: Dynamic Adaptive Hierarchical Data Aggregation for Clustered Wireless Sensor Networks. Wirel. Pers. Commun. 97, 6369–6399. https://doi.org/10.1007/s11277-017-4843-6

Razfar, M., Castro, J., Labonte, L., Rezaei, R., Ghabrial, F., Shankar, P., Besnard, E., Abedi, A., 2013. Wireless network design and analysis for real time control of launch vehicles, in: IEEE International Conference on Wireless for Space and Extreme Environments. Presented at the IEEE International Conference on Wireless for Space and Extreme Environments, pp. 1–2. https://doi.org/10.1109/WiSEE.2013.6737574

Riker, A., Cerqueira, E., Curado, M., Monteiro, E., 2016. A Two-Tier Adaptive Data Aggregation Approach for M2M Group-Communication. IEEE Sens. J. 16, 823–835. https://doi.org/10.1109/JSEN.2015.2487445

Rodriguez-Zurrunero, R., Tirado-Andrés, F., Araujo, A., 2018. YetiOS: an Adaptive Operating System for Wireless Sensor Networks, in: 2018 IEEE 43rd Conference on Local Computer Networks Workshops (LCN Workshops). Presented at the 2018 IEEE 43rd Conference on Local Computer Networks Workshops (LCN Workshops), pp. 16–22. https://doi.org/10.1109/LCNW.2018.8628500

Rosadi, I., Sakti, S.P., 2017. Low-cost wireless sensor network for small area in a building, in: 2017 International Seminar on Sensors, Instrumentation, Measurement and

Metrology (ISSIMM). Presented at the 2017 International Seminar on Sensors, Instrumentation, Measurement and Metrology (ISSIMM), pp. 115–118. https://doi.org/10.1109/ISSIMM.2017.8124273

Sahoo, J., Cherkaoui, S., Hafid, A., Sahu, P.K., 2017. Dynamic Hierarchical Aggregation for Vehicular Sensing. IEEE Trans. Intell. Transp. Syst. 18, 2539–2556. https://doi.org/10.1109/TITS.2017.2650991

Salunke, P., Kate, J., 2017. Advanced smart sensor interface in internet of things for water quality monitoring, in: 2017 International Conference on Data Management, Analytics and Innovation (ICDMAI). Presented at the 2017 International Conference on Data Management, Analytics and Innovation (ICDMAI), pp. 298–302. https://doi.org/10.1109/ICDMAI.2017.8073529

Saputra, F.A., Rasyid, M.U.H.A., Abiantoro, B.A., 2017. Prototype of early fire detection system for home monitoring based on Wireless Sensor Network, in: 2017 International Electronics Symposium on Engineering Technology and Applications (IES-ETA). Presented at the 2017 International Electronics Symposium on Engineering Technology and Applications (IES-ETA), pp. 39–44. https://doi.org/10.1109/ELECSYM.2017.8240373

Sasirekha, S., Swamynathan, S., 2017. Cluster-chain mobile agent routing algorithm for efficient data aggregation in wireless sensor network. J. Commun. Netw. 19, 392–401. https://doi.org/10.1109/JCN.2017.000063

Sasirekha, S., Swamynathan, S., 2015. A Comparative Study and Analysis of Data Aggregation Techniques in WSN. Indian J. Sci. Technol. 8. https://doi.org/10.17485/ijst/2015/v8i26/81713

Sayyad, A., Shojafar, M., Delkhah, Z., Mohammad Reza Meybodi, 2010. Improving Directed Diffusion in sensor network using learning automata: DDLA new approach in Directed Diffusion, in: 2010 2nd International Conference on Computer Technology and Development. Presented at the 2010 2nd International Conference on Computer Technology and Development, pp. 189–194. https://doi.org/10.1109/ICCTD.2010.5645888

Shi, J., Sha, M., 2019. Parameter Self-Configuration and Self-Adaptation in Industrial Wireless Sensor-Actuator Networks, in: IEEE INFOCOM 2019 - IEEE Conference on Computer Communications. Presented at the IEEE INFOCOM 2019 - IEEE

Conference on Computer Communications, pp. 658–666. https://doi.org/10.1109/INFOCOM.2019.8737467

Shin, J., Suh, C., 2011. CREEC: Chain routing with even energy consumption. J. Commun. Netw. 13, 17–25. https://doi.org/10.1109/JCN.2011.6157247

Sirsikar, S., Anavatti, S., 2015. Issues of Data Aggregation Methods in Wireless Sensor Network: A Survey. Procedia Comput. Sci., Proceedings of 4th International Conference on Advances in Computing, Communication and Control (ICAC3'15) 49, 194–201. https://doi.org/10.1016/j.procs.2015.04.244

Srisooksai, T., Keamarungsi, K., Lamsrichan, P., Araki, K., 2012. Practical data compression in wireless sensor networks: A survey. J. Netw. Comput. Appl., Collaborative Computing and Applications 35, 37–59. https://doi.org/10.1016/j.jnca.2011.03.001

Sun, C., Cembrano, G., Puig, V., Meseguer, J., 2018. Cyber-Physical Systems for Real-Time Management in the Urban Water Cycle, in: 2018 International Workshop on Cyber-Physical Systems for Smart Water Networks (CySWater). Presented at the 2018 International Workshop on Cyber-physical Systems for Smart Water Networks (CySWater), pp. 5–8. https://doi.org/10.1109/CySWater.2018.00008

Sun, Y., Song, H., Jara, A.J., Bie, R., 2016. Internet of Things and Big Data Analytics for Smart and Connected Communities. IEEE Access 4, 766–773. https://doi.org/10.1109/ACCESS.2016.2529723

Tilak, S., Abu-Ghazaleh, N.B., Heinzelman, W., 2002. A Taxonomy of Wireless Micro-Sensor Network Models. Acm Mob. Comput. Commun. Rev. 6, 28–36.

Tripathi, M., Gaur, M., Laxmi, V., Babu, B., 2013. Energy efficient LEACH-C protocol for Wireless Sensor Network. pp. 402–405. https://doi.org/10.1049/cp.2013.2620

Tseng, C.-L., Lin, F.J., 2018. Extending scalability of IoT/M2M platforms with Fog computing, in: 2018 IEEE 4th World Forum on Internet of Things (WF-IoT). Presented at the 2018 IEEE 4th World Forum on Internet of Things (WF-IoT), pp. 825–830. https://doi.org/10.1109/WF-IoT.2018.8355143

Villas, L.A., Boukerche, A., Ramos, H.S., Oliveira, H.A.B.F. de, Araujo, R.B. de, Loureiro, A.A.F., 2013. DRINA: A Lightweight and Reliable Routing Approach for In-Network Aggregation in Wireless Sensor Networks. IEEE Trans. Comput. 62, 676–689. https://doi.org/10.1109/TC.2012.31

Virmani, D., Sharma, T., Sharma, R., 2013. Adaptive Energy Aware Data Aggregation Tree for Wireless Sensor Networks. ArXiv13020965 Cs.

Wang, W., Wang, B., Liu, Z., Guo, L., Xiong, W., 2011. A Cluster-based and Tree-based Power Efficient Data Collection and Aggregation Protocol for Wireless Sensor Networks - SciAlert Responsive Version [WWW Document]. https://doi.org/10.3923/itj.2011.557.564

Wang, X., Zhou, Q., Tong, J., 2019. V-Matrix-Based Scalable Data Aggregation Scheme in WSN. IEEE Access 7, 56081–56094. https://doi.org/10.1109/ACCESS.2019.2913396

Weingartner, E., vom Lehn, H., Wehrle, K., 2009. A Performance Comparison of Recent Network Simulators, in: 2009 IEEE International Conference on Communications. Presented at the 2009 IEEE International Conference on Communications, pp. 1–5. https://doi.org/10.1109/ICC.2009.5198657

Wu, P., Wang, M., 2018. Routing Algorithm based on Energy and Hop Number for Linear Distributed WSN, in: 2018 IEEE 7th Data Driven Control and Learning Systems Conference (DDCLS). Presented at the 2018 IEEE 7th Data Driven Control and Learning Systems Conference (DDCLS), pp. 194–198. https://doi.org/10.1109/DDCLS.2018.8515912

Xinhua Liu, Fangmin Li, Hailan Kuang, Xuehong Wu, 2006. The Study of Directed Diffusion Routing Protocol Based on Clustering for Wireless Sensor Network, in: 2006 6th World Congress on Intelligent Control and Automation. Presented at the 2006 6th World Congress on Intelligent Control and Automation, pp. 5120–5124. https://doi.org/10.1109/WCICA.2006.1713366

Yang, L., Zhu, H., Zhu, Z., Luo, X., Qian, H., 2019. Distributed Ordering Transmissions for Latency-Sensitive Estimation in Wireless Sensor Networks, in: 2019 IEEE 90th Vehicular Technology Conference (VTC2019-Fall). Presented at the 2019 IEEE 90th Vehicular Technology Conference (VTC2019-Fall), pp. 1–5. https://doi.org/10.1109/VTCFall.2019.8891536

Yang, M., 2017. Data aggregation algorithm for wireless sensor network based on time prediction, in: 2017 IEEE 3rd Information Technology and Mechatronics Engineering Conference (ITOEC). Presented at the 2017 IEEE 3rd Information Technology and Mechatronics Engineering Conference (ITOEC), IEEE, Chongqing, pp. 863–867. https://doi.org/10.1109/ITOEC.2017.8122476

Yassein, M.B., Aljawarneh, S., Al-huthaifi, R.K., 2017. Enhancements of LEACH protocol: Security and open issues, in: 2017 International Conference on Engineering and Technology (ICET). Presented at the 2017 International Conference on Engineering and Technology (ICET), pp. 1–8. https://doi.org/10.1109/ICEngTechnol.2017.8308164

Yi, S., Heo, J., Cho, Y., Hong, J., 2007. PEACH: Power-efficient and adaptive clustering hierarchy protocol for wireless sensor networks. Comput. Commun., Network Coverage and Routing Schemes for Wireless Sensor Networks 30, 2842–2852. https://doi.org/10.1016/j.comcom.2007.05.034

Yıldırım, G., Tatar, Y., 2017. On WSN heterogeneity in IoT and CPSs, in: 2017 International Conference on Computer Science and Engineering (UBMK). Presented at the 2017 International Conference on Computer Science and Engineering (UBMK), pp. 1020–1024. https://doi.org/10.1109/UBMK.2017.8093421

Yıldız, H.U., Tavlı, B., Kahjogh, B.O., 2017. Assessment of wireless sensor network lifetime reduction due to elimination of critical node sets, in: 2017 25th Signal Processing and Communications Applications Conference (SIU). Presented at the 2017 25th Signal Processing and Communications Applications Conference (SIU), pp. 1–4. https://doi.org/10.1109/SIU.2017.7960228

Younis, O., Fahmy, S., 2005. An experimental study of routing and data aggregation in sensor networks. https://doi.org/10.1109/MAHSS.2005.1542773

Younis, O., Fahmy, S., 2004a. Distributed clustering in ad-hoc sensor networks: a hybrid, energy-efficient approach, in: IEEE INFOCOM 2004. Presented at the IEEE INFOCOM 2004, p. 640. https://doi.org/10.1109/INFCOM.2004.1354534

Younis, O., Fahmy, S., 2004b. HEED: a hybrid, energy-efficient, distributed clustering approach for ad hoc sensor networks. IEEE Trans. Mob. Comput. 3, 366–379. https://doi.org/10.1109/TMC.2004.41

Zhang, H., Guo, J., 2017. Application of manet routing protocol in vehicular ad hoc network based on NS3, in: 2017 7th IEEE International Conference on Electronics Information and Emergency Communication (ICEIEC). Presented at the 2017 7th IEEE International Conference on Electronics Information and Emergency Communication (ICEIEC), pp. 391–394. https://doi.org/10.1109/ICEIEC.2017.8076589

Zhang, J., Hu, P., Xie, F., Long, J., He, A., 2018. An Energy Efficient and Reliable In-Network Data Aggregation Scheme for WSN. IEEE Access 6, 71857–71870. https://doi.org/10.1109/ACCESS.2018.2882210

Zhang, L., 2018. Modeling Methods for Cloud Based Cyber Physical Systems, in: 2018 IEEE SmartWorld, Ubiquitous Intelligence Computing, Advanced Trusted Computing, Scalable Computing Communications, Cloud Big Data Computing, Internet of People and Smart City Innovation (SmartWorld/SCALCOM/UIC/ATC/CBDCom/IOP/SCI). Presented at the 2018 IEEE SmartWorld, Ubiquitous Intelligence Computing, Advanced Trusted Computing, Scalable Computing Communications, Cloud Big Data Computing, Internet of People and Smart City Innovation (SmartWorld/SCALCOM/UIC/ATC/CBDCom/IOP/SCI), pp. 1271–1276. https://doi.org/10.1109/SmartWorld.2018.00221

Zhang, Xiu, Zhang, Xin, Han, L., 2019. An Energy Efficient Internet of Things Network Using Restart Artificial Bee Colony and Wireless Power Transfer. IEEE Access 7, 12686–12695. https://doi.org/10.1109/ACCESS.2019.2892798

Zhang, Y., Ren, J., Liu, J., Xu, C., Guo, H., Liu, Y., 2017. A Survey on Emerging Computing Paradigms for Big Data. Chin. J. Electron. 26, 1–12. https://doi.org/10.1049/cje.2016.11.016

Zhou, J., Chen, S., Wang, Y., Lu, W., 2017. Spatial compression scheme for improving the lifetime of wireless sensor networks, in: 2017 IEEE 17th International Conference on Communication Technology (ICCT). Presented at the 2017 IEEE 17th International Conference on Communication Technology (ICCT), pp. 922–926. https://doi.org/10.1109/ICCT.2017.8359769

Zhu, Y., Xie, L., Yuan, T., 2012. Monitoring system for forest fire based on wireless sensor network, in: Proceedings of the 10th World Congress on Intelligent Control and Automation. Presented at the Proceedings of the 10th World Congress on Intelligent Control and Automation, pp. 4245–4248. https://doi.org/10.1109/WCICA.2012.6359191

Zikria, Y.B., Kim, S.W., Hahm, O., Afzal, M.K., Aalsalem, M.Y., 2019. Internet of Things (IoT) Operating Systems Management: Opportunities, Challenges, and Solution. Sensors 19, 1793. https://doi.org/10.3390/s19081793

Zotero, 2018. Zotero | Your personal research assistant [WWW Document]. URL https://www.zotero.org/ (accessed 3.22.20).